Numer. Math. (2009) 112:565–600
DOI 10.1007/s00211-009-0218-6

Numerische
Mathematik

Domain decomposition based $\mathcal{H}$-LU preconditioning

Lars Grasedyck · **Ronald Kriemann** ·
Sabine Le Borne

Received: 9 February 2007 / Revised: 2 December 2008 / Published online: 3 March 2009

Abstract Hierarchical matrices provide a data-sparse way to approximate fully populated matrices. The two basic steps in the construction of an $\mathcal{H}$-matrix are (a) the hierarchical construction of a matrix block partition, and (b) the blockwise approximation of matrix data by low rank matrices. In this paper, we develop a new approach to construct the necessary partition based on domain decomposition. Compared to standard geometric bisection based $\mathcal{H}$-matrices, this new approach yields $\mathcal{H}$-LU factorizations of finite element stiffness matrices with significantly improved storage and computational complexity requirements. These rigorously proven and numerically verified improvements result from an $\mathcal{H}$-matrix block structure which is naturally suited for parallelization and in which large subblocks of the stiffness matrix remain zero in an LU factorization. We provide numerical results in which a domain decomposition based $\mathcal{H}$-LU factorization is used as a preconditioner in the iterative solution of the discrete (three-dimensional) convection-diffusion equation.

Mathematics Subject Classification (2000) 65F05 · 65F30 · 65F50 · 65N55

This work was supported in part by the US Department of Energy under Grant No. DE-FG02-04ER25649 and by the National Science Foundation under grant No. DMS-0408950.

L. Grasedyck (✉) · R. Kriemann
Max-Planck-Institute for Mathematics in the Sciences,
Inselstrasse 22–26, 04103 Leipzig, Germany
e-mail: lgr@mis.mpg.de

R. Kriemann
e-mail: rok@mis.mpg.de

S. Le Borne
Tennessee Technological University,
Cookeville, TN 38505, USA
e-mail: sleborne@tntech.edu

1 Introduction

Hierarchical (or $\mathcal{H}$-) matrices have first been introduced in 1999 [12] and since then have entered into a wide range of applications. They provide a format for the data-sparse representation of fully populated matrices. The key idea is to approximate certain subblocks of a matrix by data-sparse low-rank matrices which are represented by a product of two rectangular matrices as follows: Let $A \in \mathbb{R}^{n\times n}$ with $\operatorname{rank}(A) = k$ and $k \ll n$. Then there exist matrices $B, C \in \mathbb{R}^{n\times k}$ such that $A = BC^T$. Whereas A has n^2 entries, B and C together have $2kn$ entries which results in significant savings in storage if $k \ll n$. A new $\mathcal{H}$-matrix arithmetic has been developed which allows (approximate) matrix-vector multiplication and matrix-matrix operations such as addition, multiplication, inversion and LU factorization of matrices in this format in nearly optimal complexity $\mathcal{O}(n \log^\alpha n)$ with a moderate parameter α [9]. It is even possible to reach optimal complexity $\mathcal{O}(n)$ by use of $\mathcal{H}^2$-matrices [3,16].

In finite element methods, the stiffness matrix is sparse but its inverse or LU factors are fully populated and can be approximated by $\mathcal{H}$-matrices. Such an approximate inverse, or approximate LU factors, may then be used as a preconditioner in iterative methods [19] or for the representation of matrix valued functions [5]. Even though the complexities of the $\mathcal{H}$-matrix inversion and $\mathcal{H}$-LU-factorization are of almost optimal order, there are relatively large constants involved in these complexities which in the past have prevented $\mathcal{H}$-matrix based preconditioners to be competitive with other state-of-the-art methods. In this paper, we investigate a clustering strategy which reduces the constants in the complexity estimates for the $\mathcal{H}$-LU factorization significantly: we will introduce (recursive) domain decompositions with an interior boundary, also known as *nested dissection*, into the construction of the index cluster tree of an $\mathcal{H}$-matrix. This new clustering algorithm, first presented in [20], will yield a block structure in which large subblocks are zero and remain zero in a subsequent LU factorization. As a result, the constants in the (nearly optimal) storage and work complexities will be significantly smaller than for the standard $\mathcal{H}$-matrix setting based on bisection clustering. A related approach that combines nested dissection with $\mathcal{H}$-matrix techniques has also been pursued in [18].

The clustering method can also be performed in a purely algebraic fashion by only using the sparsity pattern of a matrix [11]. For parallelization issues see also [11]. A comparison of the algebraic $\mathcal{H}$-LU preconditioner with standard direct and iterative solvers is presented in [10].

The remainder of this paper is structured as follows: Sect. 2 is devoted to preliminaries. It will provide an introduction of the model partial differential equation, a review of the nested dissection method, as well as a brief introduction to the construction and arithmetic of $\mathcal{H}$-matrices, including the $\mathcal{H}$-LU factorization for arbitrary hierarchical block clusterings. Section 3 introduces the new clustering algorithm which is based on domain decomposition. In Sect. 4 we prove that certain subblocks of the LU factors can be approximated by low rank, where the rank depends only polylogarithmically on the desired accuracy. Section 5 is devoted to complexity estimates for the storage requirement and computation of the domain decomposition based $\mathcal{H}$-LU factors. Section 6 provides numerical results for the new approach in comparison with standard bisection based $\mathcal{H}$-matrix techniques when applied to three-dimensional convection-diffusion problems.

2 Preliminaries: model problem, nested dissection and $\mathcal{H}$-matrices

2.1 The finite element model problem

Throughout this paper, we consider a linear system of the form $Au = b$, where A is the sparse Galerkin stiffness matrix of an invertible second order uniformly elliptic partial differential operator $\mathcal{A} : H_0^1(\Omega) \to H^{-1}(\Omega)$,

$$\mathcal{A}u = -\mathrm{div}\sigma\nabla u + b \cdot \nabla u + cu, \tag{1}$$

on a domain $\Omega \subset \mathbb{R}^d$ with L^∞-coefficients $\sigma : \Omega \to \mathbb{R}^{d\times d}$, $b : \Omega \to \mathbb{R}^d$, $c : \Omega \to \mathbb{R}$. The N-dimensional finite element space is denoted by $V_N \subset H_0^1(\Omega)$ and is spanned by a local basis $(\varphi_i)_{i\in\mathcal{I}}$ with index set $\mathcal{I} := \{1, \dots, N\}$, where the term "local" is defined as follows:

Assumption 1 (*Locality*) We assume that the supports of the basis functions $(\varphi_i)_{i\in\mathcal{I}}$ are locally separated in the sense that there exist two constants C_{sep} and $n_{\min}$ so that

$$\max_{i\in\mathcal{I}} \# \left\{ j \in \mathcal{I} \middle| \mathrm{dist}(\mathrm{supp}\varphi_i, \mathrm{supp}\varphi_j) \le \frac{\mathrm{diam}(\mathrm{supp}\varphi_i)}{C_{\mathrm{sep}}} \right\} \le n_{\min}. \tag{2}$$

The left-hand side is the maximal number of basis functions with "relatively close" supports.

Remark 2 1. The stiffness matrix A is sparse with at most $Nn_{\min}$ non-zero entries.
2. The locality condition (2) does not require shape regularity or a K-mesh property (neighbored elements are of comparable size). On the other hand, it bounds the number of non-neighbored elements that are close to each other in $\mathbb{R}^d$.

2.2 A review of nested dissection

Most direct methods for sparse linear systems perform an LU factorization of the original matrix after some reordering of the indices in order to reduce fill-ins. A popular reordering method is the so-called *nested dissection* method which exploits the concept of separation. The idea of nested dissection has been introduced more than 30 years ago [6] and since then attracted considerable attention (see, e.g., [4,17] and the references therein). The main idea is to separate the vertices in a (matrix) graph into three parts, two of which have no coupling between each other. The third one, referred to as an interior boundary or separator, contains couplings with (possibly both of) the other two parts. The nodes of the separated parts are numbered first and the nodes of the separator are numbered last. This process is then repeated recursively in each subgraph. An illustration of the resulting sparsity pattern is shown in Fig. 1 for

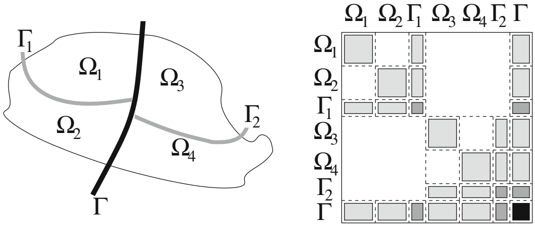


Fig. 1 Nested dissection and resulting matrix sparsity structure

the first two decomposition steps. In domain decomposition terminology, we recursively subdivide the domain into an interior boundary and the resulting two disjoint subdomains.

A favorable property of such an ordering is that a subsequent LU factorization maintains a major part of this sparsity structure, i.e., there occurs no fill-in in the large, off-diagonal zero matrix blocks that contain the coupling between two disjoint subdomains (white blocks in Fig. 1). In fact, in the case of a regular three-dimensional grid, the computational complexity amounts to $\mathcal{O}(N^2)$ for a matrix $A \in \mathbb{R}^{N\times N}$ [21]. In order to obtain a (nearly) optimal complexity, we propose to approximate the nonzero, off-diagonal blocks in the $\mathcal{H}$-matrix representation and compute them using $\mathcal{H}$-matrix arithmetic which will be introduced in Sect. 2.3. The small blocks on the diagonal and their LU factorizations will be stored as full matrices.

2.3 A brief introduction to $\mathcal{H}$-matrices

In this section, we will introduce hierarchical ($\mathcal{H}$-)matrices and their arithmetic, including the computation of an approximate $\mathcal{H}$-LU factorization. An $\mathcal{H}$-matrix provides a data-sparse approximation to a dense matrix by replacing certain blocks of the matrix by matrices of low rank which can be stored very efficiently. The blocks which allow for such low rank representations are selected from a hierarchy of partitions organized in a so-called cluster tree.

Definition 3 (*Cluster tree*) Let $T_{\mathcal{I}} = (V, E)$ be a tree with vertex set V and edge set E. For a vertex $v \in V$, we define the set of successors (or sons) of v as $S(v) := \{w \in V \mid (v, w) \in E\}$.

The tree $T_{\mathcal{I}}$ is called a cluster tree of $\mathcal{I}$ if its vertices consist of subsets of $\mathcal{I}$ and satisfy the following conditions (cf. Fig. 2, left):

1. $\mathcal{I} \in V$ is the root of $T_{\mathcal{I}}$, and $v \subset \mathcal{I}$, $v \neq \emptyset$, for all $v \in V$.
2. For all $v \in V$, there holds $S(v) = \emptyset$ or $v = \dot{\bigcup}_{w\in S(v)} w$.

The depth of a cluster tree, $d(T_{\mathcal{I}})$, is defined as the length of the longest path in $T_{\mathcal{I}}$. In the following, we identify V and $T_{\mathcal{I}}$, i.e., we write $v \in T_{\mathcal{I}}$ instead of $v \in V$. The nodes $v \in V$ are called clusters. The nodes with no successors are called *leaves* and define the set $\mathcal{L}(T_{\mathcal{I}}) := \{v \in T_{\mathcal{I}} \mid S(v) = \emptyset\}$. For a cluster $v \in T_{\mathcal{I}}$, we denote the restriction of $T_{\mathcal{I}}$ to v with vertices $\{w \in V \mid w \subset v\}$ by T_v.

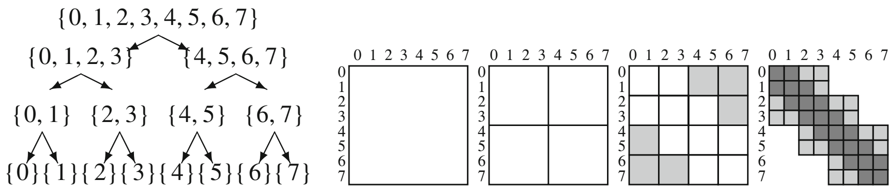


Fig. 2 *Left* A cluster tree $T_{\mathcal{I}}$. *Right* The four levels of the block cluster tree $T_{\mathcal{I}\times\mathcal{I}}$, where nodes that are further refined are *white*, inadmissible leaves are *dark grey*, and admissible leaves are *light grey*. Leaves of the block cluster tree are present on several levels of the tree

The construction of a suitable cluster tree typically considers the cardinalities and/or the geometries of the resulting clusters. We thus need the following geometric entities associated with the indices:

Definition 4 (*Geometric entities*) Every index $i \in \mathcal{I}$ is associated with a basis function φ_i of the underlying finite element space V_N. For every i, we assign a (fixed) nodal point x_i such that

$$x_i \in \operatorname{supp}\varphi_i. \tag{3}$$

For a cluster v of indices, we define its support by

$$\Omega_v := \bigcup_{j\in v} \operatorname{supp}\varphi_j. \tag{4}$$

Later, we will need (upper bounds of) the diameters of these clusters as well as the distances between two such clusters (both in the Euclidean norm, see (6,7)). Since diameters and distances can be computed much more efficiently for rectangular boxes than for arbitrarily shaped domains, we supply each cluster v with a bounding box

$$B_v = \bigotimes_{j=1}^{d} [\alpha_{v,j}, \beta_{v,j}] \tag{5}$$

that contains Ω_v, i.e., $\Omega_v \subset B_v$.

Given a cluster tree $T_{\mathcal{I}}$, any two clusters $s, t \in T_{\mathcal{I}}$ form a product $s \times t$, also called a *block cluster*, which can be associated with the corresponding matrix block $(A_{ij})_{i\in s, j\in t}$. We will use an admissibility condition to decide whether such a block will be allowed in a block partition of the matrix A or should be further refined. In general, an admissibility condition is a boolean function

$$\text{Adm} : T_{\mathcal{I}} \times T_{\mathcal{I}} \to \{\texttt{true}, \texttt{false}\}.$$

Most of the previous $\mathcal{H}$-matrix papers (e.g., [8,9,14,15]) employ the *standard* (or *strong*) admissibility condition which is given by

$$\begin{aligned} \mathrm{Adm}_{\mathrm{S}}(s \times t) &= \mathtt{true} \\ &:\Leftrightarrow \min(\mathrm{diam}(B_s), \mathrm{diam}(B_t)) \le \eta \,\mathrm{dist}(B_s, B_t) \end{aligned} \tag{6}$$

for some $0 < \eta$. Here, B_s, B_t are the bounding boxes (5) of the clusters s, t, respectively.

In actual computations, a weaker admissibility condition achieves considerable savings in both storage and work complexities while maintaining a sufficient approximation accuracy. Such a weak admissibility is given by

$$\begin{aligned} \mathrm{Adm}_{\mathrm{W}}(s \times t) &= \mathtt{true} \\ &:\Leftrightarrow \#\{i \in \mathcal{I} \mid \mathrm{supp}\varphi_i \cap B_s \cap B_t \neq \emptyset\} \le n_{\min}. \end{aligned} \tag{7}$$

The condition (7) means that blocks are admissible if they overlap at most for a few basis functions.

Given a cluster tree $T_{\mathcal{I}}$ and an admissibility condition, we construct a hierarchy of block partitionings of the product index set $\mathcal{I} \times \mathcal{I}$. The hierarchy forms a tree structure and is organized in the *block cluster tree* $T_{\mathcal{I}\times\mathcal{I}}$:

Definition 5 (*Block cluster tree*) Let $T_{\mathcal{I}}$ be a cluster tree of the index set $\mathcal{I}$. A cluster tree $T_{\mathcal{I}\times\mathcal{I}}$ is called a block cluster tree (based upon $T_{\mathcal{I}}$) if for all $v \in T_{\mathcal{I}\times\mathcal{I}}$ there exist $s, t \in T_{\mathcal{I}}$ such that $v = s \times t$. The nodes $v \in T_{\mathcal{I}\times\mathcal{I}}$ are called block clusters.

A block cluster tree may be constructed from a given cluster tree in the following canonical way which is also employed for all subsequent block cluster trees in this paper.

Construction 6 (Canonical block cluster tree construction) *Given a cluster tree* $T_{\mathcal{I}}$*, an admissibility condition* $Adm(\cdot)$*, and a parameter* $n_{\min}$*, we construct a block cluster tree* $T_{\mathcal{I}\times\mathcal{I}}$ *by*

$$\mathrm{root}(T_{\mathcal{I}\times\mathcal{I}}) := \mathcal{I} \times \mathcal{I},$$

and each vertex $s \times t \in T_{\mathcal{I}\times\mathcal{I}}$ *has the set of successors*

$$S(s \times t) := \begin{cases} \emptyset & \textit{if}\ \mathrm{Adm}(s \times t) = \mathtt{true}; \\ \emptyset & \textit{if}\ \min\{\#s, \#t\} \le n_{\min}; \\ \{s' \times t' \mid s' \in S(s), t' \in S(t)\} & \textit{otherwise.} \end{cases} \tag{8}$$

The parameter $n_{\min}$ (from Assumption 1) has to be chosen large enough to fulfill (2). For rather small blocks the matrix arithmetic of a full matrix is more efficient than that of a structured matrix. Therefore, $n_{\min}$ should be chosen at least $n_{\min} \ge 10$, which is typically at the same time sufficient for Assumption 1.

In Fig. 2, we have provided a simple example for a cluster tree and the corresponding block cluster tree. The indices in this example correspond to the continuous, piecewise linear basis functions of a regularly refined unit interval (in lexicographical order). Matrix blocks which correspond to admissible block clusters will be approximated in a data-sparse format by the following Rk-matrix representation.

Definition 7 (*Rk-matrix representation*) Let $k, n, m \in \mathbb{N}_0$. Let $M \in \mathbb{R}^{n \times m}$ be a matrix of at most rank k. A representation of M in factorized form

$$M = AB^T, \qquad A \in \mathbb{R}^{n \times k}, \quad B \in \mathbb{R}^{m \times k} \tag{9}$$

with A and B stored in full matrix representation, is called an Rk-matrix representation of M, or, in short, we call M an Rk-matrix.

If the rank k is small compared to the matrix size given by n and m, we obtain considerable savings in the storage and work complexities of an Rk-matrix compared to a full matrix [9]. Finally, we can introduce the following definition of a hierarchical matrix:

Definition 8 ($\mathcal{H}$*-matrix*) Let $k, n_{\min} \in \mathbb{N}_0$. The set of $\mathcal{H}$-matrices induced by a block cluster tree $T := T_{\mathcal{I} \times \mathcal{I}}$ with blockwise rank k and minimum block size $n_{\min}$ is defined by

$$\mathcal{H}(T, k) := \left\{ M \in \mathbb{R}^{\mathcal{I} \times \mathcal{I}} \middle| \forall s \times t \in \mathcal{L}(T) : \operatorname{rank}(M|_{s \times t}) \leq k \right.$$
$$\left. \text{or } \min\{\#s, \#t\} \leq n_{\min} \right\}.$$

Blocks $M|_{s \times t}$ with $\operatorname{rank}(M|_{s \times t}) \leq k$ are stored as Rk-matrices whereas all other blocks are stored as full matrices.

Whereas the classical $\mathcal{H}$-matrix uses a fixed rank for the Rk-blocks, it is possible to replace it by *variable* (*or adaptive*) *ranks* in order to enforce a desired relative accuracy within the individual blocks [9].

2.4 Arithmetic of $\mathcal{H}$-matrices

Given two $\mathcal{H}$-matrices $A, B \in \mathcal{H}(T, k)$ based on the same block cluster tree T, i.e., with the same block structure, the exact sum or product of these two matrices will typically not belong to $\mathcal{H}(T, k)$. In the case of matrix addition, we have $A + B \in \mathcal{H}(T, 2k)$; the rank of an exact matrix product is less obvious. We will use a truncation operator $\mathcal{T}^{\mathcal{H}}_{k \leftarrow k'}$ to define the $\mathcal{H}$-matrix addition $C := A \oplus_{\mathcal{H}} B$ and $\mathcal{H}$-matrix multiplication $C := A \otimes_{\mathcal{H}} B$ such that $C \in \mathcal{H}(T, k)$.

A truncation $\mathcal{T}_{k \leftarrow k'}(R)$ of a rank k' matrix R to rank k is defined as a best approximation with respect to the Frobenius (or spectral) norm in the set of rank k matrices. In the context of $\mathcal{H}$-matrices, we use such truncations for all admissible (rank k') blocks. Using truncated versions of the QR-decomposition and singular value decomposition, the truncation of a rank k' matrix $R \in \mathbb{R}^{n,m}$ (given in the form $R = AB^T$ where $A \in \mathbb{R}^{n,k'}$ and $B \in \mathbb{R}^{m,k'}$) to a lower rank can be computed with complexity $\mathcal{O}\left((k')^2(n + m)\right)$; further details are provided in [9]. We then define the $\mathcal{H}$-matrix addition and multiplication as follows:

$$A \oplus_{\mathcal{H}} B := \mathcal{T}^{\mathcal{H}}_{k \leftarrow 2k}(A + B); \quad A \otimes_{\mathcal{H}} B := \mathcal{T}^{\mathcal{H}}_{k \leftarrow k'}(A \cdot B),$$

where $k' \leq c(p+1)k$ is the rank of the exact matrix product, c denotes some constant (which depends on the block cluster tree T) and p denotes the depth of the tree, cf. Definition 3. Estimates that show that the $\mathcal{H}$-matrix addition and multiplication have almost optimal complexity are provided in [9] along with details on the efficient implementation of these operations.

The approximate $\mathcal{H}$-matrix addition and multiplication permit the explicit computation of an approximate LU factorization in $\mathcal{H}$-matrix format (a so-called $\mathcal{H}$-LU factorization). Such a factorization is defined recursively in the block structure of the $\mathcal{H}$-matrix. If A corresponds to a leaf block $s \times t$ on the matrix diagonal, then the exact LU factorization $A = LU$ is computed. If, however, A is further refined, i.e., $S(s \times t) = \{s_i \times t_j \mid i, j = 1, \ldots, n\}$, then a (block) $\mathcal{H}$-LU factorization

$$\begin{pmatrix} A_{1,1} & \cdots & A_{1,n} \\ \vdots & \ddots & \vdots \\ A_{n,1} & \cdots & A_{n,n} \end{pmatrix} = \begin{pmatrix} L_{1,1} & & 0 \\ \vdots & \ddots & \\ L_{n,1} & \cdots & L_{n,n} \end{pmatrix} \begin{pmatrix} U_{1,1} & \cdots & U_{1,n} \\ & \ddots & \vdots \\ 0 & & U_{n,n} \end{pmatrix}$$

is computed as in Algorithm 1 (where we abbreviate the submatrix $A|_{s_i \times t_j}$ by $A_{i,j}$).

Algorithm 1 $\mathcal{H}$-LU factorization

```
for $i = 1, \cdots, n$ do
    for $j = 1, \cdots, i-1$ do
        Solve $\sum_{k=1}^{j} L_{i,k} U_{k,j} = A_{i,j}$ for $L_{i,j}$;                (*)
    endfor;
    Compute $L_{i,i}, U_{i,i}$ as an $\mathcal{H}$-LU factorization:
        $L_{i,i} U_{i,i} = A_{i,i} - \sum_{k=1}^{i-1} L_{i,k} U_{k,i}$;                (**)
    for $j = i+1, \cdots, n$ do
        Solve $\sum_{k=1}^{i} L_{i,k} U_{k,j} = A_{i,j}$ for $U_{i,j}$;                (***)
    endfor;
endfor;
```

The solves for $L_{i,j}$ and $U_{i,j}$ in (*) and (***) of Algorithm 1 require lower and upper triangular solves in $\mathcal{H}$-arithmetic, respectively. These triangular solves are once again defined recursively in the $\mathcal{H}$-matrix block structure where on the coarsest (i.e., no further refined) level, an exact triangular solve is computed in exact (full) matrix arithmetic. We note that the summation and products $\sum L_{i,k} U_{k,j}$ of matrices in (*), (**) and (***) of Algorithm 1 are not performed exactly but in $\mathcal{H}$-arithmetic.

3 Domain decomposition based clustering

The storage and computational complexities and also the accuracy of an $\mathcal{H}$-LU factorization $A \approx L^{\mathcal{H}} U^{\mathcal{H}}$ depend strongly on the $\mathcal{H}$-matrix block structure which in turn strongly depends on the underlying cluster tree construction. Our goal is to derive a clustering strategy that will yield an $\mathcal{H}$-matrix block structure which is better suited for the computation of the $\mathcal{H}$-LU factors of a (stiffness) matrix than the standard

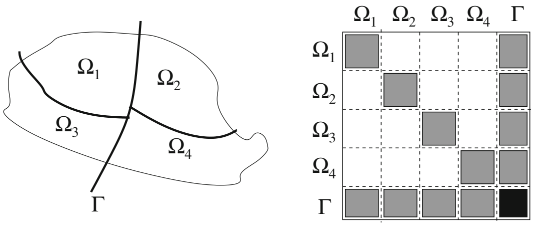


Fig. 3 Direct domain decomposition and the resulting matrix sparsity structure

geometric bisection clustering. We derive a new algorithm to construct a cluster tree which will permit a subsequent $\mathcal{H}$-LU factorization in which

- large off-diagonal blocks remain zero,
- non-zero off-diagonal blocks can be approximated in $\mathcal{H}$-matrix format, and
- the factorization process is well-suited for parallelization.

The new clustering algorithm is based on a domain decomposition approach. In [13], a direct domain decomposition method has been combined with the hierarchical matrix technique. In particular, a domain Ω is subdivided into p subdomains and an interior boundary Γ which separates the subdomains as shown in Fig. 3. Within each subdomain, standard $\mathcal{H}$-matrix techniques are used, i.e., $\mathcal{H}$-matrices are constructed by the standard bisection index clustering with zero or two successors as will be explained in Sect. 3.1. Thus, the *first* step is the decomposition of the domain into a fixed number of subdomains, and in a *second* step, $\mathcal{H}$-matrix techniques are applied within each subdomain. The new approach of this paper is to combine (or unify) these two steps: we will take the index clustering of the domain decomposition and use this same clustering as a cluster tree for the $\mathcal{H}$-matrix construction.

3.1 Clustering based on bisection

The standard construction of the cluster tree $T_{\mathcal{I}}$ is based on variants of binary space partitioning. The basic idea to construct the clusters is to subdivide a cluster v with support Ω_v (4) into smaller clusters v_1, v_2 as follows:

1. Let Q_v denote a box that contains all nodal points $(x_i)_{i\in v}$, cf. (3). For the root cluster this could be the bounding box $Q_{\mathcal{I}} := B_{\mathcal{I}}$.
2. Subdivide the box Q_v into two parts $Q_v = Q_1 \dot{\cup} Q_2$ of equal size.
3. Define the two sons $S(v) = \{v_1, v_2\}$ of v by

$$v_1 := \{i \in v \mid x_i \in Q_1\}, \quad v_2 := \{i \in v \mid x_i \in Q_2\}$$

   and use the boxes $Q_{v_1} := Q_1$, $Q_{v_2} := Q_2$ for the further splitting of the sons.

The subdivision is typically performed such that the resulting diameters of the boxes associated with successor clusters become as small as possible so that clusters eventually become separated and fulfill the standard admissibility condition (6). A visualization of this geometric regular bisection process is given in Fig. 4.

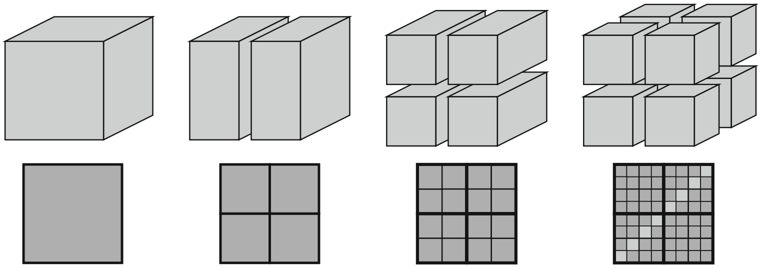


Fig. 4 The box $Q_{\mathcal{I}}$ (*top left*) that contains the whole domain $\Omega = [0, 1]^3$ is successively subdivided into two subboxes. The corresponding matrix A bears a hierarchical 2×2 block-structure. The *eight light grey blocks* are not admissible with respect to (6) but weakly admissible (since they only "touch" in one corner)

3.2 Clustering based on domain decomposition

A new construction of the cluster tree is based on the decomposition of a cluster v with corresponding domain Ω_v (4) into three sons, i.e., $S(v) = \{v_1, v_2, v_3\}$, where v_1 and v_2 contain the indices of the two disconnected subdomains Ω_{v_1} and Ω_{v_2} while $v_3 = v \backslash (v_1 \cup v_2)$ contains the indices corresponding to the separator $\Gamma_v = \Omega_{v_3}$, cf. Fig. 1. The formal DD-clustering process is presented in the following Construction 9.

Visualizations of the clustering process and the resulting block structures for two- and three-dimensional sample domains are presented in Figs. 5 and 6, respectively.

Construction 9 (DD-clustering) *The cluster tree $T_{\mathcal{I}}$ as well as the set of domain-clusters $\mathcal{C}_{\mathrm{dom}}$ and interface-clusters $T_{\mathcal{I}} \backslash \mathcal{C}_{\mathrm{dom}}$ are constructed recursively, starting with*

- *the root $\mathcal{I}$, $\mathcal{C}_{\mathrm{dom}} := \{\mathcal{I}\}$ and*
- *the box $Q_{\mathcal{I}} := B_{\mathcal{I}}$ from (5) that contains the domain Ω.*

Clusters that satisfy $\#v \le n_{\min}$ will be no further refined. For all other clusters, we distinguish between domain-clusters and interface-clusters. If a cluster v and a box

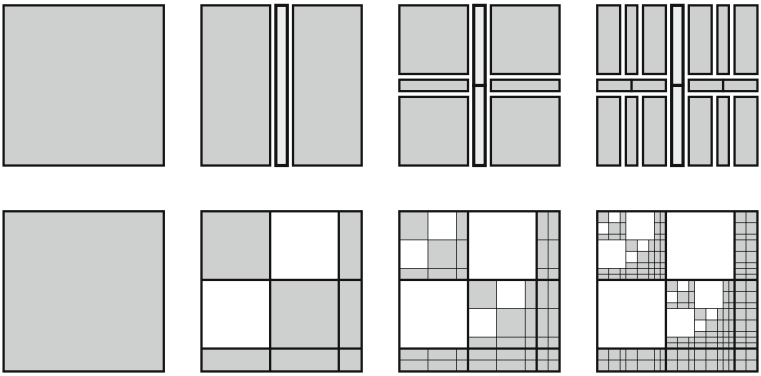


Fig. 5 *Top row* the box $Q_{\mathcal{I}}$ (*top left*) that contains the whole domain $\Omega = [0, 1]^2$ is successively subdivided by DD-clustering. The interface cluster (*bold*) is subdivided by standard bisection in every other step. *Bottom row* the four levels of the corresponding block cluster tree

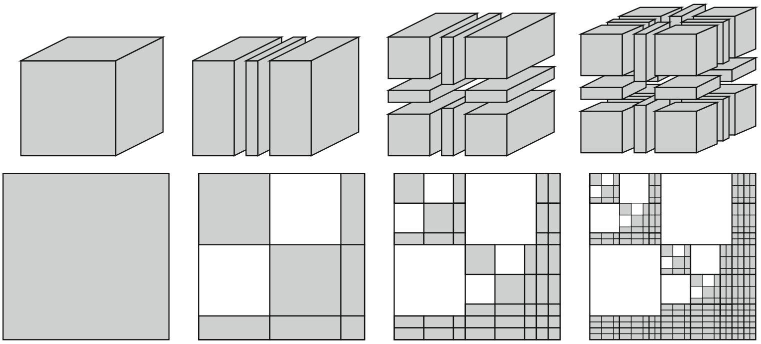


Fig. 6 The domain $\Omega = [0, 1]^3$ is successively subdivided using DD-clustering. The *top row* displays the respective bounding boxes used for the subdivision. The *bottom row* shows the resulting hierarchical block structure of the matrix A. The *white* blocks $s \times t$ satisfy $\mathrm{Adm}_{\mathrm{DD}}(s \times t) = \texttt{true}$

$Q_v = \bigotimes_{i=1}^{d} [\alpha_i, \beta_i]$ satisfying $x_j \in Q_v$ for all $j \in v$ are given, we introduce new boxes Q_1 and Q_2 by splitting Q_v in half in the coordinate direction i_{split} of maximal extent. In the following, we use the notation

$$X_i := \mathrm{supp}\varphi_i.$$

Domain-clusters: *For $v \in \mathcal{C}_{\mathrm{dom}}$, we define the three successors (cf. Fig. 5, top row)*

$$\begin{aligned} v_1 &:= \{i \in v \mid x_i \in Q_1\}, \quad v_2 := \{i \in v \mid X_i \cap \Omega_{v_1} = \emptyset\}, \\ v_3 &:= v \backslash (v_1 \cup v_2). \end{aligned} \tag{10}$$

We set $S(v) := \{v_1, v_2, v_3\}$. We add the new clusters v_1, v_2 to the set of domain-clusters, i.e., $\mathcal{C}_{\mathrm{dom}} := \mathcal{C}_{\mathrm{dom}} \cup \{v_1, v_2\}$, and we associate the boxes $Q_{v_1} := Q_1$ and $Q_{v_2} := Q_2$ with v_1, v_2, resp. The interface-cluster v_3 is equipped with the "flat" box $Q_{v_3} := \bigotimes_{i=1}^{d} [\tilde{\alpha}_i, \tilde{\beta}_i]$ where $\tilde{\alpha}_i := \alpha_i$ and $\tilde{\beta}_i := \beta_i$ except for the splitting coordinate $i = i_{\mathrm{split}}$ where we set

$$\tilde{\alpha}_i := \frac{\alpha_i + \beta_i}{2} - h_{v_3}, \quad \tilde{\beta}_i := \frac{\alpha_i + \beta_i}{2} + h_{v_3}, \quad h_{v_3} := \max_{j \in v_3} \mathrm{diam}(X_j).$$

Interface-clusters: *We define the interface-level of a cluster, $\mathrm{level}_{\mathrm{int}}(v)$, as the distance of v to the nearest domain-cluster in the cluster tree. To subdivide an interface cluster $v \in T_{\mathcal{I}} \backslash \mathcal{C}_{\mathrm{dom}}$ with associated flat box Q_v, we split Q_v into two boxes $Q_v = Q_1 \dot{\cup} Q_2$ in a direction j different from the flat direction i_{split}. More precisely, the set of sons $S(v)$ is defined by*

$$S(v) := \begin{cases} \{v\} & \textit{if } \mathrm{level}_{\mathrm{int}}(v) \equiv 0 (\mathrm{mod}\ d), \\ \{\{i \in v \mid x_i \in Q_1\}, \ \{i \in v \mid x_i \in Q_2\}\} & \textit{otherwise.} \end{cases} \tag{11}$$

The associated boxes of the sons are $Q_{v_i} := Q_i$.

Remark 10 (Ordered index set) The ordering of the indices in an LU factorization is essential for the resulting work and storage requirements. As indicated above, the three sons v_1, v_2, v_3 of a *domain-cluster* v are assumed to be ordered so that the domain-clusters v_1, v_2 come first and the interface-cluster v_3 last, i.e.,

$$\max_{i \in v_1} i < \min_{j \in v_2} j \le \max_{j \in v_2} j < \min_{l \in v_3} l.$$

The two sons of an *interface-cluster* v are given in any fixed order, without loss of generality

$$\max_{i \in v_1} i < \min_{j \in v_2} j.$$

An example of a typical cluster tree for a two-dimensional problem is given in Fig. 5 (top row). In the first subdivision step the sons $S(\mathcal{I}) = \{v_1, v_2, v_3\}$ are created. The set of domain-clusters is thus $\mathcal{C}_{\text{dom}} = \{\mathcal{I}, v_1, v_2\}$. The distance of v_3 to the nearest domain-cluster (which is $\mathcal{I}$) in the tree is $\text{level}_{\text{int}}(v_3) = 1$. Therefore, in the second subdivision step the interface-cluster v_3 is split into two sons $S(v_3) = \{v_{3_1}, v_{3_2}\}$. The two sons of v_3 fulfil $2 = \text{level}_{\text{int}}(v_{3_i}) \equiv 0 (\text{mod } 2)$ and will not be subdivided in the third subdivision step, cf. (11).

Remark 11 1. The construction of the cluster tree is guided by the fact that matrix entries A_{ij} equal zero if the corresponding supports of basis functions are disjoint. One can therefore replace the condition "$X_i \cap \Omega_{v_1} = \emptyset$" in (10) by "$A_{ij} = 0$ for all $j \in v_1$".

2. The subdivision of interface-clusters $v \in T_{\mathcal{I}} \backslash \mathcal{C}_{\text{dom}}$ is delayed every dth step in order to calibrate the diameters of interface-clusters with those of domain-clusters: the cardinality of a domain cluster $v_{\text{dom},\ell}$ on level ℓ of the cluster tree is roughly $N/2^\ell$ and the diameter is roughly $\text{diam}(\Omega)/2^{\ell/d}$. On the other hand, without the delay in every the dth step, the cardinality of an interface cluster $v_{\text{intf},\ell}$ (successor of the first interface on level 1) is roughly $N^{1-1/d}/2^\ell$ and the diameter is roughly

$$\text{diam}(v_{\text{intf},\ell}) \approx \text{diam}(\Omega)/2^{\ell/(d-1)} \ll \text{diam}(\Omega)/2^{\ell/d} \approx \text{diam}(v_{\text{dom},\ell}).$$

For example, on level $\ell = \sqrt{2}\log_2(N)$ of a cluster tree based on the domain $\Omega = [0,1]^2$, there are domain-clusters of cardinality $\sqrt{N}$ and interface-clusters of cardinality 1. The ratio between the diameters is roughly $N^{\sqrt{2}/2}$, so that there are relatively large domain-clusters surrounded by many very small interface-clusters. This imbalance would lead to undesirable fill-in in the $\mathcal{H}$-matrix so that we would loose the almost linear complexity.

The canonical Construction 6 of a block cluster tree from a cluster tree requires an admissibility condition. Due to the fact that the distance between subdomain clusters Ω_{v_1} and Ω_{v_2}, given by the width of the separator Γ_v, is typically very small compared to the diameters of $\Omega_{v_1}, \Omega_{v_2}$, the block cluster $v_1 \times v_2$ is not admissible with respect to the strong admissibility condition (6). However, since the corresponding matrix block is zero and remains zero during an LU factorization, we want to regard it as admissible; in fact, we can assign a fixed rank of zero to this block. This means that an admissibility condition suitable for domain decomposition based block clusters has to distinguish between the sets of

- domain-clusters $\mathcal{C}_{\text{dom}}$ and
- interface-clusters $T_{\mathcal{I}} \backslash \mathcal{C}_{\text{dom}}$

as specified in Construction 9.

Definition 12 (*DD-admissibility*) Let $T_{\mathcal{I}}$ be a cluster tree for the index set $\mathcal{I}$ and let $\mathcal{C}_{\text{dom}} \subset T_{\mathcal{I}}$ be the subset of domain-clusters as defined in Construction 9. We define the DD-admissibility condition by

$$\text{Adm}_{\text{DD}}(s \times t) = \texttt{true} \quad :\Leftrightarrow \quad \begin{cases} (s \neq t, \;\; s, t \in \mathcal{C}_{\text{dom}}) \text{ or} \\ \text{Adm}_{\mathcal{S}}(s \times t) = \texttt{true}, \end{cases} \tag{12}$$

where $\text{Adm}_{\mathcal{S}}$ denotes the strong admissibility (6). A weak DD-admissibility is defined by replacing $\text{Adm}_{\mathcal{S}}$ by $\text{Adm}_{\mathcal{W}}$ (7) in (12).

A visualization of the DD-clustering is given in Fig. 6 for the first three subdivision steps (top). The corresponding block cluster tree and partition of the matrix (bottom) is done by the canonical Construction 6. The characteristic sparsity pattern can be noticed already after the first subdivision step.

The $\mathcal{H}$-LU factorization given in Algorithm 1 simplifies as follows for a matrix block $A_{s \times s}$: If s is a further refined interface cluster, i.e., $s \in T_{\mathcal{I}} \backslash \mathcal{C}_{\text{dom}}$, there holds $n = 2$ since interface clusters are refined by bisection. If, however, s is a further refined domain-cluster, i.e., $s \in \mathcal{C}_{\text{dom}}$, there holds $n = 3$ with matrix blocks $A_{1,2} = A_{2,1} = 0$ as illustrated in Fig. 6. As a result, there holds $U_{1,2} = L_{2,1} = 0$ in the LU-factorization so that the $\mathcal{H}$-LU Algorithm 1 simplifies to four triangular solves ($U_{1,3}$, $U_{2,3}$, $L_{3,1}$, $L_{3,2}$, resp.) and three $\mathcal{H}$-LU factorizations on the next coarser level ($L_{1,1}U_{1,1}$, $L_{2,2}U_{2,2}$, $L_{3,3}U_{3,3}$, resp.).

4 Existence of approximate $\mathcal{H}$-LU factors

In this section, we state the important result that the $\mathcal{H}$-matrix format defined by DD-clustering allows to approximate the exact LU-factors with an approximation error ε and a blockwise rank k_{LU} that depends polylogarithmically on ε.

Our analysis is based on two previous results: The existence proof in [2] establishes the approximation of the inverse of a finite element stiffness matrix by an $\mathcal{H}$-matrix up to an accuracy of the finite element error. In this proof, the $\mathcal{H}$-matrix is based on standard bisection-clustering, and a blockwise rank that depends polylogarithmically on the accuracy is shown to be sufficient. Numerical results, however, show that in fact an arbitrary accuracy can be reached, i.e., no limitation by the finite element accuracy can be observed. Also, we observe that a blockwise rank

$$k_{\text{inv}} \sim |\log \varepsilon|^{d-1}$$

is sufficient to reach the accuracy

$$\|A^{-1} - A_{\mathcal{H}}^{-1}\|_2 \leq \varepsilon.$$

In [1], it is proved that there exist $\mathcal{H}$-LU factors $L_{\mathcal{H}}, U_{\mathcal{H}}$ that essentially yield the same approximation quality as the $\mathcal{H}$-inverse $A_{\mathcal{H}}^{-1}$ but with a blockwise rank of

$$k_{\mathrm{LU}} \lesssim k_{\mathrm{inv}} \log^{\beta}(\mathrm{cond}_2(A)),$$

again for $\mathcal{H}$-matrices based on the standard (strong) admissibility condition and standard bisection-clustering. The techniques that we employ for the proof are quite similar, but we are able to generalize this result to $\mathcal{H}$-matrices based on DD-clustering, and we establish the quasi-optimal rank bound

$$k_{\mathrm{LU}} \lesssim k_{\mathrm{inv}}.$$

In fact, the numerical results suggest that

$$k_{\mathrm{LU}} \sim |\log \varepsilon|^{d-2}.$$

In summary, for a given accuracy ϵ there exist $\mathcal{H}$-matrices $L_{\mathcal{H}}$ and $U_{\mathcal{H}}$ such that $\|A - L_{\mathcal{H}}U_{\mathcal{H}}\| \leq \epsilon$, both for the case of geometric bisection (proven in [1]) and domain decomposition based clustering (proven here). Furthermore, the storage and computational complexities of the $\mathcal{H}$-LU factorization are almost linear in the problem size N, cf. Sect. 5.

The general idea for the proof is as follows:

1. It is shown that all Schur complements in A (in particular Schur complements with respect to admissible blocks) have efficient $\mathcal{H}$-matrix approximations (Theorem 15). Note that the inverse cannot be approximated efficiently by an $\mathcal{H}$-matrix based on DD-clustering since the large, off-diagonal zero matrix blocks would not remain zero during the inversion.
2. We provide a recursion formula for the admissible blocks of the exact LU factors based on generalized Schur complements.
3. We prove that the submatrices of L and U corresponding to admissible blocks can be approximated by Rk-matrices with the same rank as the respective submatrices of an $\mathcal{H}$-matrix approximation of the Schur complement (Theorem 24).

The main existence result is stated in Theorem 24 following a technical proof of an auxiliary Lemma.

Notation 13 We denote the restriction of the block cluster tree T to $s \times t \subset \mathcal{I} \times \mathcal{I}$ and its successors by $T|_{s\times t}$.

We assume that the inverse of each minor B of A (possibly but not necessarily a FEM matrix) can be approximated by an $\mathcal{H}$-matrix $B_{\mathcal{H}}^{-1} \in \mathcal{H}(T, k_{\mathrm{inv}})$, where the block cluster tree T has been constructed by the canonical Construction 6 using a domain decomposition based cluster tree and the strong admissibility condition (6). This means that all admissible blocks are admissible in the classical sense and not with respect to the DD-admissibility condition. We cannot use the DD-admissibility here, because the zero-blocks would be of (almost) full rank in the inverse. However, the approximate inverse is, for the theory, only required in intermediate steps.

Assumption 14 (*Existence of an $\mathcal{H}$-matrix inverse*) For any $\varepsilon > 0$ and $r := \{1, \ldots, n\}$, $n \leq N$, the minor $B := A|_{r\times r}$ is invertible, and there exists an $\mathcal{H}$-matrix $B_{\mathcal{H}}^{-1} \in \mathcal{H}(T|_{r\times r}, k_{\text{inv}})$ with

$$k_{\text{inv}} := (\log n)^2 |\log \varepsilon|^{d+1} \quad \text{and} \quad \|B^{-1} - B_{\mathcal{H}}^{-1}\|_2 \leq C_{\text{inv}}\varepsilon.$$

Given Assumption 14, we now prove the existence of $\mathcal{H}$-LU factors for domain-decomposition based $\mathcal{H}$-matrix structures following the steps outlined above. For this purpose, we define a general Schur complement $S(s, t)$ with respect to the matrix block $s \times t$ (cf. Fig. 7) by

$$S(s, t) := A|_{s\times t} - A|_{s\times r}(A|_{r\times r})^{-1}A|_{r\times t}, \tag{13}$$

where $r := \{i \in \mathcal{I} \mid i < \min\{j \in s \cup t\}\}$ (see also Remark 10 concerning the ordering).

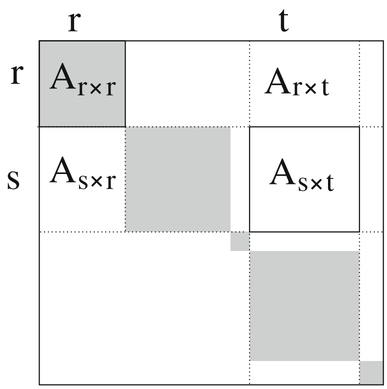
Theorem 15 (Approximation of Schur complements) *Let $A \in \mathcal{H}(T, k_{\text{inv}})$ (k_{inv} from Assumption 14), and let $b = s \times t$ be any block cluster. Then the Schur complement*

$$S(s, t) = A|_{s\times t} - A|_{s\times r}(A|_{r\times r})^{-1}A|_{r\times t}$$

of the block b in A can be approximated by an $\mathcal{H}$-matrix $S_{\mathcal{H}}(s, t) \in \mathcal{H}(T|_{s\times t}, k')$ where $k' \lesssim (p + 1)^2 k_{\text{inv}}$, $p := \text{depth}(T_{\mathcal{I}})$, such that

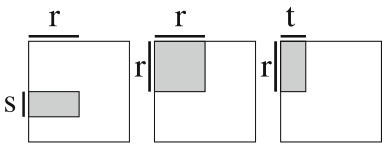
$$\|S(s, t) - S_{\mathcal{H}}(s, t)\|_2 < C_{\text{inv}}\|A\|_2^2\varepsilon.$$

Fig. 7 The Schur complement $S(s, t)$ is defined via the inverse of the diagonal block $A|_{r\times r}$ and the coupling matrices $A|_{s\times r}$ and $A|_{r\times t}$. The diagonal blocks are *grey*



Proof We define the $\mathcal{H}$-matrices $A^{s,r}, A^{r,r}, A^{r,t} \in \mathcal{H}(T, k_{\text{inv}})$ to be zero in all subblocks except

$$A^{s,r}|_{s\times r} := A|_{s\times r},\quad A^{r,r}|_{r\times r} := (A|_{r\times r})^{-1}_{\mathcal{H}},\quad A^{r,t}|_{r\times t} := A|_{r\times t}.$$



The two matrices $A^{s,r}$ and $A^{r,t}$ belong to $\mathcal{H}(T, k_{\text{inv}})$ since $A \in \mathcal{H}(T, k_{\text{inv}})$. The matrix $(A|_{r\times r})^{-1}_{\mathcal{H}}$ is the $\mathcal{H}$-matrix approximation of $(A|_{r\times r})^{-1}$ in the set $\mathcal{H}(T|_{r\times r}, k_{\text{inv}})$ and fulfills by Assumption 14 the estimate

$$\|(A|_{r\times r})^{-1} - (A|_{r\times r})^{-1}_{\mathcal{H}}\|_2 \leq C_{\text{inv}}\varepsilon.$$

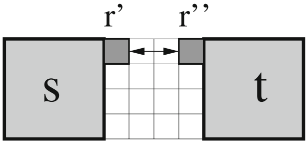
The exact product yields the proposed error estimate:

$$\|A|_{s\times r}(A|_{r\times r})^{-1}A|_{r\times t} - (A^{s,r}A^{r,r}A^{r,t})|_{s\times t}\|_2 \leq \|A\|_2 C_{\text{inv}}\varepsilon\|A\|_2.$$

The blockwise rank of the exact product $A^{s,r}A^{r,r}A^{r,t}$ is at most $\mathcal{O}((p+1)^2 k_{\text{inv}})$, since for each multiplication the blockwise rank increases by at most a factor $\mathcal{O}(p+1)$ [9, Theorem 2.24]. The submatrix $S_{\mathcal{H}}(s,t) := (A - A^{s,r}A^{r,r}A^{r,t})|_{s\times t}$ has a blockwise rank of at most k'. □

Remark 16 (a) For any domain-cluster $t \in \mathcal{C}_{\text{dom}}$, there holds $S(t,t) = A|_{t\times t}$.

(b) For an admissible block cluster $s \times t$ and a matrix $A \in \mathcal{H}(T, 0)$, one can omit the factor $(p+1)^2$ so that $k' \lesssim k_{\text{inv}}$:



Let P denote a partition of r. We write

$$A|_{s\times r}(A|_{r\times r})^{-1}A|_{r\times t} = \sum_{r'\in P}\sum_{r''\in P} A|_{s\times r'}((A|_{r\times r})^{-1})|_{r'\times r''}A|_{r''\times t}$$

and observe that an addend is non-zero only if both $s \times r'$ and $r'' \times t$ are inadmissible. Since s and t are well-separated (admissible), then $r' \times r''$ will also be admissible if the diameters of r' and r'' are by a factor smaller than the distance between s and t, i.e., $\max\{\text{diam}(r'), \text{diam}(r'')\} \leq c\,\text{dist}(s,t\}$. Therefore $((A|_{r\times r})^{-1})|_{r'\times r''}$ is of rank at most k_{inv} and thus $k' \leq (\#P)^2 k_{\text{inv}} \lesssim k_{\text{inv}}$.

In the second step of the proof, we derive formulae for the structure of the LU factors of Schur complements. The exact factors L and U will later be approximated by $\mathcal{H}$-matrices with matrix partitions as they occur in the domain decomposition based clustering. We begin with a recursion formula for Schur complements with respect to blocks on the diagonal:

Lemma 17 (Recursion formula for Schur complements I) *For the Schur complement $S(t,t)$ of an interface-cluster $t \in T_{\mathcal{I}}$ with two sons $S(t) = \{t_1, t_2\}$, there holds*

$$S(t,t) = \begin{bmatrix} S(t_1,t_1) & S(t_1,t_2) \\ S(t_2,t_1) & S(t_2,t_2) + S(t_2,t_1)S(t_1,t_1)^{-1}S(t_1,t_2) \end{bmatrix}. \tag{14}$$

If the only son of t is t_1, then $S(t,t) = S(t_1,t_1)$. For Schur complements $S(t,t)$ of domain clusters $t \in T_{\mathcal{I}}$ with three sons $S(t) = \{t_1, t_2, t_3\}$, there holds

$$S(t,t) = \begin{bmatrix} A|_{t_1\times t_1} & 0 & A|_{t_1\times t_3} \\ 0 & A|_{t_2\times t_2} & A|_{t_2\times t_3} \\ A|_{t_3\times t_1} & A|_{t_3\times t_2} & A|_{t_3\times t_3} \end{bmatrix}. \tag{15}$$

Proof We distinguish between blocks corresponding to interface-clusters and blocks corresponding to domain-clusters. An interface-cluster t (that is not a leaf) has either one son t_1 (which is the trivial case) or two sons t_1, t_2 (see (11) in Construction 9 for the domain decomposition based clustering). In the case of two sons, the corresponding diagonal matrix block is subdivided into four subblocks. For this case, the proof is given in [1, Lemma 3.1] and repeated here for completeness: Let $r := \{1, \ldots, \min\{j \in t\} - 1\}$. Then

$$\begin{aligned} S(t,t)|_{t_1\times t_1} &\overset{\text{def.}}{=} \left(A|_{t\times t} - A|_{t\times r}(A|_{r\times r})^{-1}A|_{r\times t}\right)\Big|_{t_1\times t_1} \\ &= A|_{t_1\times t_1} - A|_{t_1\times r}(A|_{r\times r})^{-1}A|_{r\times t_1} = S(t_1,t_1), \end{aligned}$$

and analogously for $S(t_1,t_2)$ and $S(t_2,t_1)$ since r in the definition of the Schur complement is the same for all three: $\min\{j \in t\} = \min\{j \in t_1\}$. For the lower right block, there holds

$$\begin{aligned} S(t_2,t_2) &= A|_{t_2\times t_2} - A|_{t_2\times r\cup t_1}(A|_{r\cup t_1\times r\cup t_1})^{-1}A|_{r\cup t_1\times t_2} \\ &= A|_{t_2\times t_2} - \begin{bmatrix} A|_{t_2\times r} & A|_{t_2\times t_1} \end{bmatrix} \begin{bmatrix} A|_{r\times r} & A|_{r\times t_1} \\ A|_{t_1\times r} & A|_{t_1\times t_1} \end{bmatrix}^{-1} \begin{bmatrix} A|_{r\times t_2} \\ A|_{t_1\times t_2} \end{bmatrix} \\ &= A|_{t_2\times t_2} - \begin{bmatrix} A|_{t_2\times r} & A|_{t_2\times t_1} \end{bmatrix} \begin{bmatrix} A|_{r\times r}^{-1} & -A|_{r\times r}^{-1}A|_{r\times t_1}S(t_1,t_1)^{-1} \\ 0 & S(t_1,t_1)^{-1} \end{bmatrix} \\ &\quad\times \begin{bmatrix} I & 0 \\ -A|_{t_1\times r}A|_{r\times r}^{-1} & I \end{bmatrix} \begin{bmatrix} A|_{r\times t_2} \\ A|_{t_1\times t_2} \end{bmatrix}. \end{aligned}$$

The product of the first two matrices equals

$$\begin{aligned}&\left[\, A|_{t_2\times r}A|_{r\times r}^{-1} \;\middle|\; (A|_{t_2\times t_1} - A|_{t_2\times r}A|_{r\times r}^{-1}A|_{r\times t_1})S(t_1,t_1)^{-1}\,\right]\\ &\quad = \left[\, A|_{t_2\times r}A|_{r\times r}^{-1} \;\middle|\; S(t_2,t_1)S(t_1,t_1)^{-1}\,\right],\end{aligned}$$

whereas the product of the last two matrices is

$$\begin{bmatrix} A|_{r\times t_2} \\ A|_{t_1\times t_2} - A|_{t_1\times r}A|_{r\times r}^{-1}A|_{r\times t_2}\end{bmatrix} = \begin{bmatrix} A|_{r\times t_2} \\ S(t_1,t_2)\end{bmatrix}.$$

Multiplying both and subtracting them from $A|_{t_2\times t_2}$ yields

$$\begin{aligned}S(t_2,t_2) &= A|_{t_2\times t_2} - A|_{t_2\times r}A|_{r\times r}^{-1}A|_{r\times t_2} - S(t_2,t_1)S(t_1,t_1)^{-1}S(t_1,t_2)\\ &= S(t,t)|_{t_2\times t_2} - S(t_2,t_1)S(t_1,t_1)^{-1}S(t_1,t_2).\end{aligned}$$

For a domain-cluster t (that is not a leaf), the Schur complement equals the matrix itself since the off-diagonal coupling matrices are zero. □

For blocks in the upper triangular part of the matrix we derive a similar recursion formula:

Lemma 18 (Recursion formula for Schur complements II) *Let $b = s\times t$ be a block in the upper triangular part, i.e.,* $\max\{j\in s\} < \min\{j\in t\}$. *Let $t'\in S(t)$ be a son of t. If s has exactly one son $S(s)=\{s'\}$, then*

$$S(s,t)|_{s'\times t'} = S(s',t').$$

If s has two successors $S(s)=\{s_1,s_2\}$, then

$$S(s,t)|_{s\times t'} = \begin{bmatrix} S(s_1,t') \\ S(s_2,t') + S(s_2,s_1)S(s_1,s_1)^{-1}S(s_1,t')\end{bmatrix}. \tag{16}$$

For a domain-cluster s with three successors $S(s)=\{s_1,s_2,s_3\}$ there holds

$$S(s,t)_{s\times t'} = \begin{bmatrix} A|_{s_1\times t'} \\ A|_{s_2\times t'} \\ A|_{s_3\times t'}\end{bmatrix}. \tag{17}$$

Proof We define $r := \{i\in\mathcal{I} \mid i < \min\{j\in s\}\}$. For $S(s)=\{s'\}$ we get

$$\begin{aligned}S(s,t)|_{s'\times t'} &= \left(A|_{s\times t} - A|_{s\times r}(A|_{r\times r})^{-1}A|_{r\times t}\right)\Big|_{s'\times t'}\\ &= A|_{s'\times t'} - A|_{s'\times r}(A|_{r\times r})^{-1}A|_{r\times t'} = S(s',t').\end{aligned}$$

For $S(s) = \{s_1, s_2\}$ we can proceed as in Lemma 17:

$$\begin{aligned} S(s,t)|_{s_1\times t'} &= \left(A|_{s\times t} - A|_{s\times r}(A|_{r\times r})^{-1}A|_{r\times t}\right)\big|_{s_1\times t'} \\ &= A|_{s_1\times t'} - A|_{s_1\times r}(A|_{r\times r})^{-1}A|_{r\times t'} \quad = \quad S(s_1,t'), \\ S(s,t)|_{s_2\times t'} &= A|_{s_2\times t'} - A|_{s_2\times r\cup s_1}(A|_{r\cup s_1\times r\cup s_1})^{-1}A|_{r\cup s_1\times t'} \\ &= A|_{s_2\times t'} - \begin{bmatrix} A|_{s_2\times r} & A|_{s_2\times s_1}\end{bmatrix} \begin{bmatrix} A|_{r\times r} & A|_{r\times s_1} \\ A|_{s_1\times r} & A|_{s_1\times s_1}\end{bmatrix}^{-1} \begin{bmatrix} A|_{r\times t'} \\ A|_{s_1\times t'}\end{bmatrix} = A|_{s_2\times t'} \\ &\quad -A|_{s_2\times r}A|_{r\times r}^{-1}A|_{r\times t'} - S(s_2,s_1)S(s_1,s_1)^{-1}S(s_1,t') \\ &= S(s,t)|_{s_2\times t'} - S(s_2,s_1)S(s_1,s_1)^{-1}S(s_1,t'). \end{aligned}$$

Formula (17) holds because s is a domain cluster and the off-diagonal coupling $A|_{s\times r}$ in the definition of $S(s,t)$ is zero. □

The following Lemma is the "transpose" of Lemma 18 and can be proven analogously.

Lemma 19 (Recursion formula for Schur complements III) *Let $b = s \times t$ be a block in the lower triangular part, i.e.,* $\min\{j \in s\} > \max\{j \in t\}$. *Let $s' \in S(s)$ be a son of s. If t has exactly one son $S(t) = \{t'\}$, then*

$$S(s,t)|_{s'\times t'} = S(s',t').$$

If t has two successors $S(t) = \{t_1, t_2\}$, then

$$S(s,t)|_{s'\times t} = \left[\, S(s',t_1) \;\middle|\; S(s',t_2) + S(s',t_1)S(t_1,t_1)^{-1}S(t_1,t_2) \,\right]. \tag{18}$$

For a domain-cluster t there holds

$$S(s,t)_{s'\times t} = \left[\, A|_{s'\times t_1} \;\middle|\; A|_{s'\times t_2} \;\middle|\; A|_{s'\times t_3} \,\right]. \tag{19}$$

In the following definition we introduce factors L and U for the representation of the Schur complement $S(t,t)$. We will later prove that these factors can be efficiently approximated by $\mathcal{H}$-matrices.

Definition 20 (*LU factors for the Schur complement*) We define the LU factors $L(t,t)$ and $U(t,t)$ for the Schur complement $S(t,t)$, $t \in T_{\mathcal{I}}$, recursively by

$$\begin{aligned} L(t,t) &:= \begin{bmatrix} L(t_1,t_1) & 0 \\ S(t_2,t_1)U(t_1,t_1)^{-1} & L(t_2,t_2)\end{bmatrix}, \\ U(t,t) &:= \begin{bmatrix} U(t_1,t_1) & L(t_1,t_1)^{-1}S(t_1,t_2) \\ 0 & U(t_2,t_2)\end{bmatrix} \end{aligned}$$

for interior interface-clusters t with sons $S(t) = \{t_1, t_2\}$, and

$$L(t,t) := L(t',t'), \quad U(t,t) := U(t',t')$$

for nodes t with only one son $S(t) = \{t'\}$. For leaves $t \in \mathcal{L}(T_\mathcal{I})$, the two factors $L(t,t)$ and $U(t,t)$ are defined as the exact LU factors of $S(t,t)$. For domain-clusters $t \in T_\mathcal{I}$ with sons $S(t) = \{t_1, t_2, t_3\}$, the factors are defined as

$$L(t,t) := \begin{bmatrix} L(t_1,t_1) & 0 & 0 \\ 0 & L(t_2,t_2) & 0 \\ A|_{t_3\times t_1}U(t_1,t_1)^{-1} & A|_{t_3\times t_2}U(t_2,t_2)^{-1} & L(t_3,t_3) \end{bmatrix} \quad \text{and}$$

$$U(t,t) := \begin{bmatrix} U(t_1,t_1) & 0 & L(t_1,t_1)^{-1}A|_{t_1\times t_3} \\ 0 & U(t_2,t_2) & L(t_2,t_2)^{-1}A|_{t_2\times t_3} \\ 0 & 0 & U(t_3,t_3) \end{bmatrix}.$$

Lemma 21 (Exact LU factorization of Schur complements) *Let $S(t,t)$ denote a Schur complement for the cluster $t \in T_\mathcal{I}$. Then the LU factors in Definition 20 fulfil*

$$L(t,t)U(t,t) = S(t,t).$$

Proof We prove the equation by induction over the depth of the block cluster tree, where the start is given by definition of the exact LU factors (depth zero). Now let t be an interior node of the cluster tree $T_\mathcal{I}$. For interface-clusters t, there holds

$$\begin{aligned}
(L(t,t)U(t,t))\,|_{t_1\times t_1} &\overset{\text{def.}}{=} L(t_1,t_1)U(t_1,t_1) \overset{\text{Ind.}}{=} S(t_1,t_1),\\
(L(t,t)U(t,t))\,|_{t_1\times t_2} &\overset{\text{def.}}{=} L(t_1,t_1)L(t_1,t_1)^{-1}S(t_1,t_2) = S(t_1,t_2),\\
(L(t,t)U(t,t))\,|_{t_2\times t_1} &\overset{\text{def.}}{=} S(t_2,t_1)U(t_1,t_1)^{-1}U(t_1,t_1) = S(t_2,t_1),\\
(L(t,t)U(t,t))\,|_{t_2\times t_2} &\overset{\text{def.}}{=} L(t_2,t_2)U(t_2,t_2)\\
&\quad +S(t_2,t_1)U(t_1,t_1)^{-1}L(t_1,t_1)^{-1}S(t_1,t_2)\\
&\overset{\text{Ind.}}{=} S(t_2,t_2) + S(t_2,t_1)S(t_1,t_1)^{-1}S(t_1,t_2).
\end{aligned}$$

According to Lemma 17 the assertion follows.

Now let t be a domain-cluster. We prove the statement $L(t,t)U(t,t) = S(t,t)$ again by induction, where the trivial start is a leaf t where the equation holds by definition. Multiplication of $L(t,t)$ and $U(t,t)$ yields

$$L(t,t)U(t,t) = \begin{bmatrix} L(t_1,t_1)U(t_1,t_1) & 0 & A|_{t_1\times t_3} \\ 0 & L(t_2,t_2)U(t_2,t_2) & A|_{t_2\times t_3} \\ A|_{t_3\times t_1} & A|_{t_3\times t_2} & S_{33} \end{bmatrix},$$

where

$$\begin{aligned}
S_{33} &= A|_{t_3\times t_1}U(t_1,t_1)^{-1}L(t_1,t_1)^{-1}A|_{t_1\times t_3}\\
&\quad +A|_{t_3\times t_2}U(t_2,t_2)^{-1}L(t_2,t_2)^{-1}A|_{t_2\times t_3}+L(t_3,t_3)U(t_3,t_3)\\
&\overset{\text{Ind.}}{=} A|_{t_3\times t_1}A|_{t_1\times t_1}^{-1}A|_{t_1\times t_3}+A|_{t_3\times t_2}A|_{t_2\times t_2}^{-1}A|_{t_2\times t_3}+S(t_3,t_3),\\
S(t_3,t_3) &\overset{\text{def.}}{=} A|_{t_3\times t_3}-A|_{t_3\times t_1\cup t_2}A|_{t_1\cup t_2\times t_1\cup t_2}^{-1}A|_{t_1\cup t_2\times t_3}\\
&= A|_{t_3\times t_3}-A|_{t_3\times t_1}A|_{t_1\times t_1}^{-1}A|_{t_1\times t_3}-A|_{t_3\times t_2}A|_{t_2\times t_2}^{-1}A|_{t_2\times t_3},\\
\Rightarrow\quad S_{33} &= A|_{t_3\times t_3}.
\end{aligned}$$

An induction for the first two diagonal subblocks and $S(t,t)=A|_{t\times t}$ completes the proof. □

Lemma 22 (Recursion formula for Schur complements IV) *Let $b=s\times t$ be a block in the upper triangular part, i.e.,* $\max\{j\in s\}<\min\{j\in t\}$*. Let $s'\in S(s)$ and $t'\in S(t)$. Then*

$$\left(L(s,s)^{-1}S(s,t)\right)\Big|_{s'\times t'}=L(s',s')^{-1}S(s',t'). \tag{20}$$

For the lower triangular part there holds

$$\left(S(s,t)U(t,t)^{-1}\right)\Big|_{s'\times t'}=S(s',t')U(t',t')^{-1}. \tag{21}$$

Proof We prove only the first part of the Lemma, since the second part follows analogously. Let $s\times t$ be a block in the upper triangular part and $s'\in S(s)$ and $t'\in S(t)$.

Trivial case $S(s)=\{s'\}$:

$$\begin{aligned}
\left(L(s,s)^{-1}S(s,t)\right)\Big|_{s'\times t'} &= \left(L(s',s')^{-1}S(s,t)\right)\Big|_{s'\times t'}\\
&\overset{\text{Lemma 18}}{=} L(s',s')^{-1}S(s,t').
\end{aligned}$$

Interface case $S(s)=\{s_1,s_2\}$: The inverse of the L-factor is

$$\begin{aligned}
L(s,s)^{-1} &\overset{\text{def.}}{=} \begin{bmatrix} L(s_1,s_1)^{-1} & 0\\ -L(s_2,s_2)^{-1}S(s_2,s_1)U(s_1,s_1)^{-1}L(s_1,s_1)^{-1} & L(s_2,s_2)^{-1}\end{bmatrix}\\
&= \begin{bmatrix} L(s_1,s_1)^{-1} & 0\\ -L(s_2,s_2)^{-1}S(s_2,s_1)S(s_1,s_1)^{-1} & L(s_2,s_2)^{-1}\end{bmatrix}.
\end{aligned}$$

In the following, we use the short notation $L_{\nu,\mu} := L(s_\nu, s_\mu)$. For $t' \in S(t)$, there holds

$$\begin{aligned}
\big(L(s,s)^{-1}S(s,t)\big)|_{s\times t'} &= \begin{bmatrix} L_{1,1}^{-1} & 0 \\ -L_{2,2}^{-1}S(s_2,s_1)S(s_1,s_1)^{-1} & L_{2,2}^{-1} \end{bmatrix} \begin{bmatrix} S(s,t)|_{s_1\times t'} \\ S(s,t)|_{s_2\times t'} \end{bmatrix} \\
&= \begin{bmatrix} L_{1,1}^{-1}S(s,t)|_{s_1\times t'} \\ L_{2,2}^{-1}S(s,t)|_{s_2\times t'} - L_{2,2}^{-1}S(s_2,s_1)S(s_1,s_1)^{-1}S(s,t)|_{s_1\times t'} \end{bmatrix} \\
&\overset{\text{Lemma 18}}{=} \begin{bmatrix} L(s_1,s_1)^{-1}S(s_1,t') \\ L(s_2,s_2)^{-1}S(s_2,t') \end{bmatrix}.
\end{aligned}$$

Domain case $S(s) = \{s_1, s_2, s_2\}$: we use the short notation

$$L_{3i} := L(s_3,s_3)^{-1}A|_{s_3\times s_i}S(s_i,s_i)^{-1}$$

for $i \in \{1,2\}$. Then the inverse of the L-factor reads

$$L(s,s)^{-1} \overset{\text{def.}}{=} \begin{bmatrix} L(s_1,s_1)^{-1} & 0 & 0 \\ 0 & L(s_2,s_2)^{-1} & 0 \\ -L_{31} & -L_{32} & L(s_3,s_3)^{-1} \end{bmatrix}.$$

Let $t' \in S(t)$. For the first two components $s_i \times t'$, $i \in \{1,2\}$, of the product, we have

$$\begin{aligned}
\Big(L(s,s)^{-1}S(s,t)\Big)|_{s_i\times t'} &= L(s_i,s_i)^{-1}S(s,t)|_{s_i\times t'} \\
&\overset{\text{Lemma 18}}{=} L(s_i,s_i)^{-1}S(s_i,t').
\end{aligned}$$

For the last component $s_3 \times t'$, we conclude

$$\begin{aligned}
&\Big(L(s,s)^{-1}S(s,t)\Big)|_{s_3\times t'} \\
&\quad= L(s_3,s_3)^{-1}S(s_3,t)|_{s_3\times t'} - L_{31}S(s_1,t)|_{s_1\times t'} - L_{32}S(s_2,t)|_{s_2\times t'} \\
&\quad\overset{\text{Lemma 18}}{=} L(s_3,s_3)^{-1}A|_{s_3\times t'} - L_{31}A|_{s_1\times t'} - L_{32}A|_{s_2\times t'} \\
&\quad= L(s_3,s_3)^{-1}A|_{s_3\times t'} - \sum_{i=1}^{2} L(s_3,s_3)^{-1}A|_{s_3\times s_i}S(s_i,s_i)^{-1}A|_{s_i\times t'} \\
&\quad= L(s_3,s_3)^{-1}\left(A|_{s_3\times t'} - \sum_{i=1}^{2} A|_{s_3\times s_i}S(s_i,s_i)^{-1}A|_{s_i\times t'}\right) \\
&\quad\overset{\text{Lemma 17}}{=} L(s_3,s_3)^{-1}S(s_3,t').
\end{aligned}$$

□

For the exact LU factors of a Schur complement S we are able to prove that they can be approximated in the $\mathcal{H}$-matrix format with the same blockwise rank as the $\mathcal{H}$-matrix approximation $S_{\mathcal{H}}$ of S, such that the difference between the exact and approximated factorization is under control. This is not a surprise, since Lemma 22 states

$$L(r,r)|_{s\times t} = L(s,s)^{-1}S(s,t)$$

for all admissible blocks $s \times t$, so that $L(r,r)|_{s\times t}$ can be approximated by a low rank matrix if and only if $S(s,t)$ can be approximated by a low rank matrix.

Lemma 23 (Approximate $\mathcal{H}$-LU factorization of Schur complements) *For any cluster $r \in T_{\mathcal{I}}$ the LU factors of the matrix $S(r,r)$ defined in Definition 20 can be approximated by $\mathcal{H}$-matrices*

$$L_{\mathcal{H}}(r,r), U_{\mathcal{H}}(r,r) \;\in\; \mathcal{H}(T|_{r\times r}, k_{\mathrm{LU}}), \quad k_{\mathrm{LU}} \lesssim (p+1)^2 k_{\mathrm{inv}}$$

(k_{inv} *the blockwise rank of the $\mathcal{H}$-matrix approximation to the inverse from Assumption* 14), *where the approximation error in each leaf $s \times t$ is bounded by*

$$\|\,(L_{\mathcal{H}}(r,r) - L(r,r))\,|_{s\times t}\|_2 \le C_{\mathrm{inv}} c_U \|A\|_2^2 \varepsilon,$$
$$\|\,(U_{\mathcal{H}}(r,r) - U(r,r))\,|_{s\times t}\|_2 \le C_{\mathrm{inv}} c_L \|A\|_2^2 \varepsilon.$$

The variable p is the depth of $T_{\mathcal{I}\times\mathcal{I}}$, C_{inv} is from Assumption 14, *and*

$$c_U := \max_{t\in T_{\mathcal{I}}} \|U(t,t)^{-1}\|_2, \quad c_L := \max_{t\in T_{\mathcal{I}}} \|L(t,t)^{-1}\|_2.$$

Proof We define the matrices $L_{\mathcal{H}}(r,r)$ and $U_{\mathcal{H}}(r,r)$ blockwise. For each inadmissible block $s \times t$ they coincide with $L(r,r)|_{s\times t}$ and $U(r,r)|_{s\times t}$ from Definition 21. For each admissible block we set

$$L_{\mathcal{H}}(r,r)|_{s\times t} := \mathcal{T}_{k_{\mathrm{LU}}}\left((L(r,r)|_{s\times t}\right), U_{\mathcal{H}}(r,r)|_{s\times t} := \mathcal{T}_{k_{\mathrm{LU}}}\left((U(r,r)|_{s\times t}\right),$$

where $k_{\mathrm{LU}} := k' \lesssim (p+1)^2 k_{\mathrm{inv}}$ is defined as required in Theorem 15, and $\mathcal{T}_{k_{\mathrm{LU}}}$ is the truncation to rank k_{LU}. We prove the error bound by induction over the level of r, where for leaves $r \times r \in T_{\mathcal{I}\times\mathcal{I}}$ the error is zero.

Interface-cluster: Let the interface-cluster r be subdivided into $\{r_1, r_2\}$. By induction, the error bound holds for the submatrices $L_{\mathcal{H}}(r_i, r_i)$ and $U_{\mathcal{H}}(r_i, r_i)$. It remains to show that the off-diagonal blocks

$$S(r_2, r_1)U(r_1, r_1)^{-1} \quad \text{and} \quad L(r_1, r_1)^{-1}S(r_1, r_2)$$

fulfill the error bound. This will be proven in the last part.

Domain-cluster: Let the domain-cluster r be subdivided into $\{r_1, r_2, r_3\}$. By induction the error bound holds for the submatrices $L_{\mathcal{H}}(r_i, r_i)$ and $U_{\mathcal{H}}(r_i, r_i)$. It remains to show that the off-diagonal blocks

$$S(r_3, r_i)U(r_i, r_i)^{-1} \quad \text{and} \quad L(r_i, r_i)^{-1}S(r_i, r_3), \quad i \in \{1, 2\},$$

fulfill the error bound (recall $S(r_3, r_i) = A|_{r_3 \times r_i}$ and $S(r_i, r_3) = A|_{r_i \times r_3}$). This will be proven next.

Off-diagonal products: For all off-diagonal products of the form $S(s, t)U(t, t)^{-1}$ in the lower diagonal part and $L(s, s)^{-1}S(s, t)$ in the upper diagonal part we have already shown in Lemma 22 that they consist blockwise of products of the same form. For a leaf $s \times t$ of $T_{\mathcal{I}\times\mathcal{I}}$ the Schur complement $S(s, t)$ can be approximated by $S_{\mathcal{H}}(s, t)$ of rank at most k_{LU} so that $\|S(s, t) - S_{\mathcal{H}}(s, t)\|_2 \le C_{\mathrm{inv}}\|A\|_2^2\varepsilon$ (Theorem 15). We conclude

$$\|S(s, t)U(t, t)^{-1} - S_{\mathcal{H}}(s, t)U(t, t)^{-1}\|_2 \le \|U(t, t)^{-1}\|C_{\mathrm{inv}}\|A\|_2^2\varepsilon.$$

Since $\mathcal{T}_{k_{\mathrm{LU}}}$ is defined as the best approximation with respect to the Euclidean norm, the above upper bound also holds for $L_{\mathcal{H}}(r, r)|_{s\times t}$. By the same arguments we get

$$\|U(r, r)|_{s\times t} - U_{\mathcal{H}}(r, r)|_{s\times t}\|_2 \le \|L(s, s)^{-1}\|C_{\mathrm{inv}}\|A\|_2^2\varepsilon.$$

We summarize that each admissible block $s \times t$ of the LU factors can be approximated by a matrix of rank k' so that the blockwise error is at most

$$\|U(r, r)|_{s\times t} - U_{\mathcal{H}}(r, r)|_{s\times t}\|_2 \le C_{\mathrm{inv}}c_L\|A\|_2^2\varepsilon,$$
$$\|L(r, r)|_{s\times t} - L_{\mathcal{H}}(r, r)|_{s\times t}\|_2 \le C_{\mathrm{inv}}c_U\|A\|_2^2\varepsilon.$$

□

Theorem 24 ($\mathcal{H}$-LU factorization of A) *Under Assumption 14, the matrix A can be approximated by lower and upper triangular $\mathcal{H}$-matrices*

$$L_{\mathcal{H}}, U_{\mathcal{H}} \in \mathcal{H}(T, k_{\mathrm{LU}}), \quad k_{\mathrm{LU}} \lesssim (p + 1)^2(\log n)^2(\log(1/\varepsilon) + \log(c_A))^{d+1},$$

where $c_A := C_{\mathrm{inv}}(c_U\|U\|_2 + c_L\|L_{\mathcal{H}}\|_2)(p + 1)\|A\|_2^2$, so that the approximation error is bounded by

$$\|A - L_{\mathcal{H}}U_{\mathcal{H}}\|_2 \le \varepsilon.$$

For reasonably small $\varepsilon \le c_A^{-1}$ and the improved estimate from Remark 16 for sparse FEM stiffness matrices A, we get the estimate

$$k_{\mathrm{LU}} \lesssim k_{\mathrm{inv}}.$$

Proof The blockwise norm estimate from Theorem 23 yields the global estimate [7, Theorem 6.2]

$$\begin{aligned}
&\|L(r,r) - L_{\mathcal{H}}(r,r)\|_2 \\
&\quad \le C_{\text{sp}} C_{\text{inv}} (p+1) \max_{s\times t \in T_{\mathcal{I}\times\mathcal{I}}} \|(L(r,r) - L_{\mathcal{H}}(r,r))|_{s\times t}\|_2 \\
&\quad \le C_{\text{sp}} C_{\text{inv}} c_U (p+1) \|A\|_2^2 \varepsilon,
\end{aligned}$$

where C_{sp} is the sparsity constant of Definition 26. Due to Theorem 15, we can find a rank

$$k_{\text{LU}} \lesssim (p+1)^2 (\log n)^2 \left|\log \frac{\varepsilon}{c_A}\right|^{d+1}$$

so that

$$\|S(s,t) - S_{\mathcal{H}}(s,t)\|_2 \le \|A\|_2^2 \varepsilon / (c_A C_{\text{sp}}).$$

We define the matrices $L_{\mathcal{H}} := L_{\mathcal{H}}(\mathcal{I},\mathcal{I})$ and $U_{\mathcal{H}} := U_{\mathcal{H}}(\mathcal{I},\mathcal{I})$. According to Theorem 23 (and the above global estimate), the exact LU factors $LU = A$ of A can be approximated with accuracy

$$\begin{aligned}
\|L - L_{\mathcal{H}}\|_2 &\le C_{\text{inv}} c_U (p+1) \|A\|_2^2 \varepsilon / c_A, \\
\|U - U_{\mathcal{H}}\|_2 &\le C_{\text{inv}} c_L (p+1) \|A\|_2^2 \varepsilon / c_A.
\end{aligned}$$

Both together yield

$$\begin{aligned}
&\|A - L_{\mathcal{H}} U_{\mathcal{H}}\|_2 \le \|(L - L_{\mathcal{H}}) U_{\mathcal{H}}\|_2 + \|L_{\mathcal{H}} (U - U_{\mathcal{H}})\|_2 \\
&\quad \le \|U\|_2 C_{\text{inv}} c_U (p+1) \|A\|_2^2 \varepsilon / c_A + \|L_{\mathcal{H}}\|_2 C_{\text{inv}} c_L (p+1) \|A\|_2^2 \varepsilon / c_A \\
&\quad = \varepsilon.
\end{aligned}$$

Using the improved estimate from Remark 16, we have $k_{\text{LU}} \lesssim (\log n)^2 |\log \frac{\varepsilon}{c_A}|^{d+1}$. For $\varepsilon \le c_A$, this simplifies to

$$k_{\text{LU}} \lesssim (\log n)^2 |\log \varepsilon^2|^{d+1} \lesssim (\log n)^2 |\log \varepsilon|^{d+1} \lesssim k_{\text{inv}}.$$

□

Corollary 25 *For a block cluster tree based on DD-clustering and the DD-admissibility, both Lemma 23 and Theorem 24 apply. Blocks that are not standard admissible but only DD-admissible remain zero during the LU factorization, and for all other blocks the approximation is proven.*

5 Complexity estimates

In this section, we will prove that the storage and computational complexities of the $\mathcal{H}$-LU factorization as computed by Algorithm 1 are almost optimal, i.e., in $\mathcal{O}(N \log^c N)$ for a moderate c, cf. Corollary 31 for the main result. An important quantity which will enter the complexity estimates and turns out to be bounded by a constant is the so-called *sparsity* of the block cluster tree $T_{\mathcal{I}\times\mathcal{I}}$ [9]:

Definition 26 (*Sparsity*) The sparsity of a block cluster tree $T_{\mathcal{I}\times\mathcal{I}}$ based on a cluster tree $T_{\mathcal{I}}$ is defined by

$$C_{\text{sp}} := \max_{s\in T_{\mathcal{I}}} \#\{t \in T_{\mathcal{I}} \mid s \times t \in T_{\mathcal{I}\times\mathcal{I}}\}.$$

For a fixed domain Ω and a local mesh (see Assumption 1), the sparsity of a block cluster tree created from a regular, geometrical bisection based cluster tree by the canonical Construction 6 using the standard admissibility (6) is bounded by a constant [9, Lemma 4.5]. Therefore, C_{sp} is called the sparsity *constant*.

For the new domain decomposition based clustering and the DD-admissibility condition (12), we will also prove that C_{sp} is bounded by a constant. In order to simplify the presentation, we assume the following:

- The subdivision in Construction 9 is modified so that each domain-cluster is split d times so that there are 2^d domain sons and $d2^{d-1}$ interface sons, while each interface-cluster is split $d-1$ times into 2^{d-1} sons.
- The initial box $Q_{\mathcal{I}} = B_{\mathcal{I}}$ for the entire domain Ω is the unit cube $[0, 1]^d$.

Lemma 27 (DD-sparsity) *Let* $h := \min_{i\in\mathcal{I}} \text{diam}(\text{supp}\varphi_i)$ *and assume that locality as expressed in* (2) *holds for some constants* $C_{\text{sep}}, n_{\min}$. *Let* $T := T_{\mathcal{I}\times\mathcal{I}}$ *be the block cluster tree constructed in the canonical way* (*cf.* (8)) *from the cluster tree* $T_{\mathcal{I}}$ *from DD-clustering in Construction* 9. *Then the following statements hold:*

(a) *The depth of the tree is bounded by*

$$\text{depth}(T) < \max\left\{1,\ \log_2\left(C_{\text{sep}}\sqrt{d}h^{-1}\right)\right\} = \mathcal{O}(\log N).$$

(b) *The sparsity is bounded by*

$$C_{\text{sp}} \le 3^d(1+d)\left(1+2\left(\sqrt{d}\left(\eta^{-1}(3+2C_{\text{sep}})+4(1+C_{\text{sep}})\right)\right)\right)^d$$
$$= \mathcal{O}(\eta^{-d}).$$

Proof We denote by level(t) the distance of a cluster t to the root $\mathcal{I}$ of the cluster tree $T_{\mathcal{I}}$.
(a) Let $t \in T_{\mathcal{I}}$ be a non-leaf node and $\ell :=$ level(t). We denote the box corresponding to t by Q_t, where due to the construction diam$(Q_t) = \sqrt{d}2^{-\ell}$. Due to Construction 9,

the size of t is at least $\#t > n_{\min}$, Therefore, by Assumption 1, there exist $i, j \in t$ such that

$$\operatorname{dist}(\operatorname{supp}\varphi_i, \operatorname{supp}\varphi_j) > C_{\text{sep}}^{-1}\operatorname{diam}(\operatorname{supp}\varphi_i)$$

and thus

$$\begin{aligned}\sqrt{d}2^{-\ell} &= \operatorname{diam}(Q_t) \geq \operatorname{dist}(\operatorname{supp}\varphi_i, \operatorname{supp}\varphi_j)\\ &> C_{\text{sep}}^{-1}\operatorname{diam}(\operatorname{supp}\varphi_i) \;\geq\; C_{\text{sep}}^{-1}h. \end{aligned} \tag{22}$$

This yields $\ell < \log_2(C_{\text{sep}}\sqrt{d}h^{-1})$ and thus the proposed bound on the depth of the tree T.

(b) To prove the bound on the sparsity, we exploit the structure of the regular subdivision of $[0, 1]^d$ into boxes Q_t as follows:

1. Let $t \in T_{\mathcal{I}}$ be a node with $\operatorname{level}(t) = \ell$ and $\#t > n_{\min}$. The number of domain-boxes Q_s on level ℓ that touch Q_t is at most 3^d. By induction, it follows that the number of domain-boxes on level ℓ with a distance less than $j2^{-\ell}$ to Q_t is bounded by $(1+2j)^d$. The number of interface-boxes (between the domain-boxes) is then bounded by $d(1+2j)^d$.
2. Let $s \in T_{\mathcal{I}}$ with $\operatorname{level}(s) = \ell$, $\#s > n_{\min}$ and $\operatorname{dist}(Q_t, Q_s) \geq j2^{-\ell}$. Using the notation $h_v := \max_{\iota \in v} \operatorname{diam}(\operatorname{supp}\varphi_\iota)$, we can estimate the diameter and distance of the respective bounding boxes B_t, B_s of the clusters by

$$\begin{aligned} \operatorname{diam}(B_t) &\leq \operatorname{diam}(Q_t) + 2h_t \overset{(22)}{\leq} \sqrt{d}2^{-\ell} + 2(1+C_{\text{sep}})\sqrt{d}2^{-\ell},\\ \operatorname{diam}(B_s) &\overset{(22)}{\leq} \sqrt{d}2^{-\ell} + 2(1+C_{\text{sep}})\sqrt{d}2^{-\ell},\\ \operatorname{dist}(B_s, B_t) &\geq \operatorname{dist}(Q_s, Q_t) - h_t - h_s\\ &\geq j2^{-\ell} - 4(1+C_{\text{sep}})\sqrt{d}2^{-\ell}. \end{aligned}$$

3. If $s \times t$ is not admissible, then the domain-decomposition admissibility (12) yields the relation

$$j2^{-\ell}\eta < \sqrt{d}2^{-\ell} + 2(1+C_{\text{sep}})\sqrt{d}2^{-\ell} + 4(1+C_{\text{sep}})\sqrt{d}2^{-\ell}\eta,$$

which gives the desired estimate

$$j < \sqrt{d}\left(\eta^{-1}(3+2C_{\text{sep}}) + 4(1+C_{\text{sep}})\right) =: j_{\max}.$$

4. As a consequence of 1. and 3., the number of nodes $s \in T_{\mathcal{I}}$ (with $\operatorname{level}(s) = \operatorname{level}(t)$, $\#s > n_{\min}$ and $\#t > n_{\min}$) not admissible to t is bounded by $(1+d)(1+2j_{\max})^d$.
5. Let $t' \in T_{\mathcal{I}}$ be arbitrary. If t' is the root of $T_{\mathcal{I}}$, then there is exactly one cluster on the same level, namely t' itself. Therefore a sparsity constant $C_{\text{sp}} \geq 1$ would be sufficient. If t' is not the root, then the father cluster t of t' fulfills $\#t > n_{\min}$. Due

to 4., we conclude that there are at most $(1+d)(1+2j_{\max})^d$ clusters $s \in T_{\mathcal{I}}$ with $s \times t \in T_{\mathcal{I}\times\mathcal{I}}$ so that there are at most $\max_{s\in T_{\mathcal{I}}} \#\mathrm{S}(s)(1+d)(1+2j_{\max})^d$ clusters $s' \in T_{\mathcal{I}}$ with $s' \times t' \in T_{\mathcal{I}\times\mathcal{I}}$. This is the desired bound for the sparsity C_{sp}.

□

The previous Lemma gives a rigorous proof that the sparsity constant C_{sp} is independent of the problem size N as well as the geometry and grows like $\mathcal{O}(\eta^{-d})$. If we include the fact that distinct domain-clusters s, t always yield admissible block clusters $s \times t$ (cf. (12)) and regard all clusters with corresponding boxes touching in at most one corner admissible, i.e., if we use the weak domain-decomposition admissibility, then there holds

$$C_{\mathrm{sp}} \approx \begin{cases} 11 & \text{if } d = 2, \\ 63 & \text{if } d = 3. \end{cases}$$

Now that we have established the bound for C_{sp}, we can apply the results from [9] in order to derive the complexity bounds for the storage and matrix-vector product for $\mathcal{H}$-matrices with constant rank k. For adaptively chosen ranks we get the same bounds by taking k as the maximum over all blockwise ranks.

Corollary 28 *Let $M, M' \in \mathcal{H}(T, k)$ for a block cluster tree $T := T_{\mathcal{I}\times\mathcal{I}}$ based on the cluster tree $T_{\mathcal{I}}$ with sparsity constant C_{sp}. Then the storage requirements $N_{\mathcal{H},St}(T, k)$, the matrix-vector complexity $N_{\mathcal{H}\cdot v}(T, k)$ and the complexity $N_{\mathcal{H}\oplus\mathcal{H}}(T, k)$ of the (formatted) matrix addition for M, M' in $\mathcal{H}$-matrix representation can be bounded by*

$$\begin{aligned} N_{\mathcal{H},St}(T,k) &\leq C_{\mathrm{sp}}(\mathrm{depth}(T)+1)\max\{k, n_{\min}\}N = \mathcal{O}(kN\log N), \\ N_{\mathcal{H}\cdot v}(T,k) &\leq 2N_{\mathcal{H},St}(T,k), \\ N_{\mathcal{H}\oplus\mathcal{H}}(T,k) &\leq 74\max\{1,k\}N_{\mathcal{H},St}(T,k). \end{aligned}$$

Proof The bounds on storage and matrix-vector multiplication have already been proven in [9, Lemma 2.4, Lemma 2.5] for arbitrary $\mathcal{H}$-matrix structures. It remains to prove the estimate for the addition which we do here in two steps. For admissible blocks, we have to truncate an $n \times m$ Rk-matrix from rank $2k$ down to k.

1. If $2k \leq \min\{n, m\}$, then we use the truncation algorithm from [9, Lemma 1.3] of complexity

$$24k^2(n+m) + 184k^3 \leq 24k^2(n+m) + 92k^2\min\{n,m\} \leq 70k^2(n+m).$$

2. If $2k > \min\{n, m\}$, then we first convert the Rk-matrix to a standard fullmatrix R of size $n \times m$ with complexity

$$4nkm \leq 4k\max\{n,m\}\min\{n,m\} < 8k^2\max\{n,m\}.$$

For the fullmatrix R, we compute a truncated QR-decomposition in complexity

$$4\max\{n,m\}\min\{n,m\}^2,$$

the SVD of the R-factor in $21\min\{n,m\}^3$ and the multiplication of the Q-factor with one of the orthogonal matrices from the SVD in $2\max\{n,m\}\min\{n,m\}^2$. All four steps sum up to

$$\begin{aligned}&8k^2\max\{n,m\}+6\max\{n,m\}\min\{n,m\}^2+21\min\{n,m\}^3\\&\quad\leq 74k^2(n+m).\end{aligned}$$

The truncation complexity of each Rk-block is at most $74k$ times the storage complexity. In the fullmatrix blocks, we have to add nm entries which is at most 1 times the number of floats to be stored. □

In order to estimate the complexity for the matrix-matrix multiplication and matrix factorization, the standard approach from [9] requires a bound on the *idempotency* of $T_{\mathcal{I}\times\mathcal{I}}$. Roughly speaking, the idempotency constant $C_{\mathrm{id}}(r\times t)$ counts the number of blocks $M|_{r'\times s'}$ and $M|_{s'\times t'}$ such that $M|_{r'\times s'}\cdot M|_{s'\times t'}$ contributes to the block $r\times t$ in the matrix product. For the block cluster tree $T_{\mathcal{I}\times\mathcal{I}}$ based on domain decomposition with the DD-admissibility (12), the standard definition of the idempotency has to be modified to exclude the admissible domain-domain block clusters since these blocks will not be modified during the factorization (but there would occur fill-in during a matrix multiplication or inversion).

Lemma 29 (DD-idempotency) *Let $T:=T_{\mathcal{I}\times\mathcal{I}}$ be the block cluster tree constructed in the canonical way* (*cf.* (8)) *from the cluster tree $T_{\mathcal{I}}$ from the DD-clustering in Construction 9. Then the idempotency constant*

$$C_{\mathrm{id}}:=\max_{r\times t\in\mathcal{L}(T),(r,t)\notin\mathcal{C}_{\mathrm{dom}}\times\mathcal{C}_{\mathrm{dom}}}C_{\mathrm{id}}(r\times t)$$

with blockwise idempotency

$$\begin{aligned}C_{\mathrm{id}}(r\times t):=\#\big\{r'\times t'\mid r'\subset r,t'\subset t\text{ and }\exists s'\in T_{\mathcal{I}}:\\ r'\times s'\in T,s'\times t'\in T\big\}\end{aligned}$$

is bounded by

$$C_{\mathrm{id}}\leq\left(\sqrt{d}(1+\eta)(3+2C_{\mathrm{sep}})\right)^{2\log_2(3)d}=\mathcal{O}(1).$$

Proof We use the notations from Lemma 27 and denote by level(t) the distance of a cluster t to the root $\mathcal{I}$ of the cluster tree $T_{\mathcal{I}}$.
Let $r\times t\in\mathcal{L}(T)$ and $(r,t)\notin\mathcal{C}_{\mathrm{dom}}\times\mathcal{C}_{\mathrm{dom}}$, $\ell:=\mathrm{level}(r)=\mathrm{level}(t)$. If $r\times t$ is not admissible, then $C_{\mathrm{id}}(r\times t)=1$ (either r or t has no sons). Now let $r\times t$ be (standard) admissible. We define $q:=\log_2\left(\sqrt{d}(1+\eta)(3+2C_{\mathrm{sep}})\right)$.

1. We will prove that for all nodes $r',s',t'\in T_{\mathcal{I}}$ with level$(r',s',t')\geq\ell+q$, $r'\times s'\subset r\times s$ and $s'\times t'\subset s\times t$, one of the vertices $r'\times s'$ or $s'\times t'$ is a leaf.

Let r', s', t' be given as above and $\min\{\#r', \#s', \#t'\} > n_{\min}$ (otherwise one of the three is a leaf of $T_{\mathcal{I}}$).
For $u \in \{r', s', t'\}$ we can bound the diameter of the corresponding bounding box as in Lemma 27 by

$$\operatorname{diam}(B_u) \leq \sqrt{d}(3 + 2C_{\mathrm{sep}})2^{-q-\ell} \overset{\mathrm{Def.}q}{\leq} 2^{-\ell}/(1+\eta). \tag{23}$$

The distance between r' and s' or s' and t' is at least

$$\begin{aligned}\max\{\operatorname{dist}(B_{r'}, B_{s'}), \operatorname{dist}(B_{s'}, B_{t'})\} &\geq \operatorname{dist}(B_{r'}, B_{t'}) - \operatorname{diam}(B_{s'})\\ &\geq \operatorname{dist}(B_r, B_t) - \operatorname{diam}(B_{s'})\\ &\overset{(6)}{\geq} \eta^{-1} \min\{\operatorname{diam}(B_r), \operatorname{diam}(B_t)\} - \operatorname{diam}(B_{s'})\\ &\geq \eta^{-1}2^{-\ell} - 2^{-\ell}/(1+\eta) = \eta^{-1}2^{\ell}/(1+\eta).\end{aligned}$$

Both estimates together yield

$$\operatorname{diam}(B_{s'}) \leq \eta \max\{\operatorname{dist}(B_{r'}, B_{s'}), \operatorname{dist}(B_{s'}, B_{t'})\},$$

so that either $r' \times s'$ or $s' \times t'$ is admissible, i.e., a leaf.

2. From (1), it follows that on a level $\geq \ell + q + 1$ there are no vertices $r' \times s' \in T$ and $s' \times t' \in T$ with $r' \subset r$, $t' \subset t$. Since the number of sons of a block cluster is limited by 3^{2d}, there are at most $3^{2dq} = 2^{q \log_2(3)2d}$ such vertices on level $\ell, \dots, \ell + q$.

□

Theorem 30 ($\mathcal{H}$-matrix multiplication) *The complexity $N_{\mathcal{H}\otimes\mathcal{H}}$ of the (formatted) matrix multiplication in $\mathcal{H}(T, k)$ with a block cluster tree T of depth $\mathcal{O}(\log N)$ and a prescribed zero-pattern where domain-domain blocks remain zero during the multiplication, is*

$$N_{\mathcal{H}\otimes\mathcal{H}}(T, k) = \mathcal{O}(k^2 N \log^2 N).$$

Proof According to [9, Theorem 2.20], the complexity to compute the exact product (without truncation) is $N_{\mathcal{H}\cdot\mathcal{H}}(T, k) = \mathcal{O}(k \log(N) N_{\mathcal{H},St}(T, k))$. The product matrix is an element of $\mathcal{H}(T, \tilde{k})$, where the blockwise rank is at most $\tilde{k} = C_{\mathrm{id}}C_{\mathrm{sp}}(\operatorname{depth}(T) + 1)k$, except for the domain-domain blocks which are enforced to remain zero. The truncation of the exact product to blockwise rank k is split into two parts. The first part concerns the domain-domain blocks. Since these are kept as zero blocks, there is nothing to be done. The second part concerns the standard admissible blocks of T. Their truncation is done by $C_{\mathrm{id}}C_{\mathrm{sp}}(\operatorname{depth}(T) + 1) - 1$ times truncation from rank $2k$ to k as in [9, Lemma 2.10]. This complexity is bounded by $74(C_{\mathrm{id}}C_{\mathrm{sp}}(\operatorname{depth}(T) + 1) - 1)kN_{\mathcal{H},St}(T, k)$. □

In principle we could use Theorem 30 in order to bound the complexity of the $\mathcal{H}$-LU factorization (based on the DD-clustering). The estimate would then read

$N_{\mathcal{H}\text{-}LU}(T,k) = \mathcal{O}(k^2 N \log^3 N)$. However, we will present a more elegant way to avoid the extra logarithm and also to reveal the improved estimate $N_{\mathcal{H}\text{-}LU}(T,k) \approx \frac{1}{2} N_{\mathcal{H}\otimes\mathcal{H}}(T,k)$.

Corollary 31 ($\mathcal{H}$-LU factorization) *For the complexity $N_{\mathcal{H}\text{-}LU}(T,k)$ of the (formatted) $\mathcal{H}$-LU factorization, there holds*

$$N_{\mathcal{H}\text{-}LU}(T,k) \le N_{\mathcal{H}\otimes\mathcal{H}}(T,k).$$

For a balanced cluster tree $T_{\mathcal{I}}$ (domain-clusters on the same level are of almost equal cardinality), there holds in particular

$$N_{\mathcal{H}\text{-}LU}(T,k) \approx \frac{1}{2} N_{\mathcal{H}\otimes\mathcal{H}}(T,k).$$

Proof Since interface clusters are refined by bisection, we will first consider a block cluster tree T based on standard geometric bisection from Sect. 3.1. We prove the statement by induction over the depth of $T_{\mathcal{I}\times\mathcal{I}}$, starting with the trivial case of a full matrix where we assume that the factorization costs approximately half of a multiplication. During the (formatted) matrix multiplication $C := A \otimes B$ of matrices $A, B, C \in \mathcal{H}(T,k)$, we have to perform the formatted operations

$$C_{ij} := A_{i1} \otimes B_{1j} \oplus A_{i2} \otimes B_{2j}, \quad i, j \in \{1, 2\}.$$

During an $\mathcal{H}$-LU factorization $C = LU$ with lower triangular factor L and upper triangular factor U (cf. Algorithm 1) we have to

1. factorize $C_{11} = L_{11}U_{11}$;
2. solve two triangular systems $L_{11}U_{12} = C_{12}$ and $L_{21}U_{11} = C_{21}$ for the unknowns U_{12}, L_{21};
3. compute the (formatted) matrix product and sum $\tilde{C}_{22} := C_{22} \ominus L_{21} \otimes U_{12}$, and finally
4. factorize $\tilde{C}_{22} = L_{22}U_{22}$ (recursion).

Steps (1) and (4) appear also during the matrix multiplications $C_{11} := A_{11} \otimes B_{11} \oplus \cdots$ and $C_{22} := A_{22} \otimes B_{22} \oplus \ldots$. By induction, they are at least of (twice for a balanced cluster tree) the complexity of the respective factorization. In step (3) we have to multiply and add $\tilde{C}_{22} := C_{22} \ominus L_{21} \otimes U_{12}$. The same operation occurs in the multiplication $C_{22} := A_{21} \otimes B_{12} \oplus \ldots$. Additionally, the multiplication $C_{11} := A_{12} \otimes B_{21} \oplus \ldots$ occurs during the multiplication, which is omitted for the LU decomposition (for a balanced cluster tree both multiplications are of almost the same complexity and therefore together twice as costly as step (3)). For the two triangular solves in (2), we have the four counterparts

$$C_{12} := A_{11} \otimes B_{12} \oplus A_{12} \otimes B_{22},$$
$$C_{21} := A_{22} \otimes B_{21} \oplus A_{21} \otimes B_{11}.$$

It remains to show that the triangular solves are of at most the complexity of the respective (formatted) matrix multiplications.

We prove the last assertion, that the multiplication $Y := L \otimes X$ is of at least the complexity of the triangular solve $LX = Y$, by induction over the depth of $T_{\mathcal{I}\times\mathcal{I}}$. For the triangular solve of

$$\begin{bmatrix} L_{11} & \\ L_{21} & L_{22} \end{bmatrix} \cdot \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix} = \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix}$$

with unknown X, we have to

1. solve 4 times a triangular system $L_{ii}X_{ij} = Y_{ij}$, $i, j \in \{1, 2\}$;
2. perform the (formatted) operation $Y_{2j} \ominus L_{21} \otimes X_{1j}$, $j \in \{1, 2\}$.

For the matrix multiplication we have to multiply $Y_{ij} := L_{ii} \otimes X_{ij}$, $i, j \in \{1, 2\}$, which is by induction at least of the complexity of the triangular solves in (1), and we perform the (formatted) operations $Y_{2j} \oplus L_{21} \otimes X_{1j}$, $j \in \{1, 2\}$, which is the same as in (2). The other two multiplications $Y_{1j} \oplus L_{12} \otimes X_{2j}$, $j \in \{1, 2\}$, are omitted in the triangular solve due to the structure of L. This completes the proof for the cluster tree $T_{\mathcal{I}}$ constructed by the standard geometric bisection from Sect. 3.1.

For the domain decomposition based cluster tree from Construction 9, the same technique as above can be applied, except that we keep the zero pattern of the matrix. The factorization

$$\begin{bmatrix} C_{11} & & C_{13} \\ & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} = \begin{bmatrix} L_{11} & & \\ & L_{22} & \\ L_{31} & L_{32} & L_{33} \end{bmatrix} \cdot \begin{bmatrix} U_{11} & & U_{13} \\ & U_{22} & U_{23} \\ & & U_{33} \end{bmatrix}$$

resolves into

1. two recursions $C_{ii} = L_{ii}U_{ii}$, $i \in \{1, 2\}$;
2. four domain-interface triangular solves $C_{3i} = L_{3i}U_{ii}$ and $C_{i3} = L_{ii}U_{i3}$ for $i \in \{1, 2\}$;
3. two matrix multiplications in $\tilde{C}_{33} := C_{33} \oplus (L_{31} \otimes U_{13}) \oplus (L_{32} \otimes U_{23})$ and
4. the factorization of the interface part $\tilde{C}_{33} = L_{33}U_{33}$.

The multiplication of two matrices A, B with zero-blocks $A_{12}, A_{21}, B_{12}, B_{21}$, requires in the non-zero blocks of C the formatted operations

1. $C_{ii} := A_{ii} \otimes B_{ii} \oplus A_{i3}B_{3i}$, $i \in \{1, 2\}$;
2. four domain-interface and interface-interface multiplications $C_{3i} := A_{3i} \otimes B_{ii} \oplus A_{33} \otimes B_{3i}$ and $C_{i3} := A_{ii} \otimes B_{i3} \oplus A_{i3} \otimes B_{33}$ for $i \in \{1, 2\}$;
3. three matrix multiplications $\tilde{C}_{33} := A_{31} \otimes B_{13} \oplus A_{32} \otimes B_{23} \oplus A_{33} \otimes B_{33}$.

By induction, parts (1) and (2) of the factorization are less costly than those of the multiplication, and part (3) of the multiplication covers the two multiplications of (3) and the recursion (4) in the factorization. This concludes the proof for non-balanced cluster trees.

For a balanced cluster tree, the triangular solves in step (2) are of half the complexity of the respective multiplications. The two extra multiplications in step (2) of the multiplication together with the two off-diagonal multiplications in step (3) are of twice the complexity as the two multiplications in step (3) of the factorization. □

6 Numerical results

In the first two examples, $\mathcal{H}$-matrices based upon geometric bisection and nested dissection clustering (cf. Fig. 8 for typical block structures) are compared for the solution of Poisson's equation

$$-\Delta u = f \quad \text{in} \quad \Omega =]0,1[^d, \quad d \in \{2,3\}. \tag{24}$$

For the discretization of (24), the finite element method with piecewise linear basis functions is used. The $\mathcal{H}$-matrices will be generated using the respective strong admissibility conditions (6), (12), with $\eta := 2$. Furthermore, the minimal cluster size is set to $n_{\min} := 20$. All computations are performed on an AMD Opteron with 2.4 GHz CPU clock rate and 8GB main memory.

Two different cases for solving (24) are considered. In the first one, an iterative method is applied and hence only a rough approximation of the LU (or rather LL^T) factors is required.

Table 1 shows the results for this computation. Along with the time for the Cholesky factorization and the size of the decomposed matrix, also the accuracy δ of the adaptive $\mathcal{H}$-matrix arithmetic is presented, which is adjusted to maintain the relative inversion error $\|I - A(LL^T)^{-1}\|_2 \sim 10^{-1}$.

The results demonstrate a significant advantage of the clustering based on nested dissection over geometric bisection. It takes less time to compute the Cholesky factorization, and the resulting $\mathcal{H}$-matrix requires less storage. One also notices that the asymptotic behavior of $\mathcal{H}$-matrix arithmetic is reached already for smaller problem sizes compared to the standard bisection method.

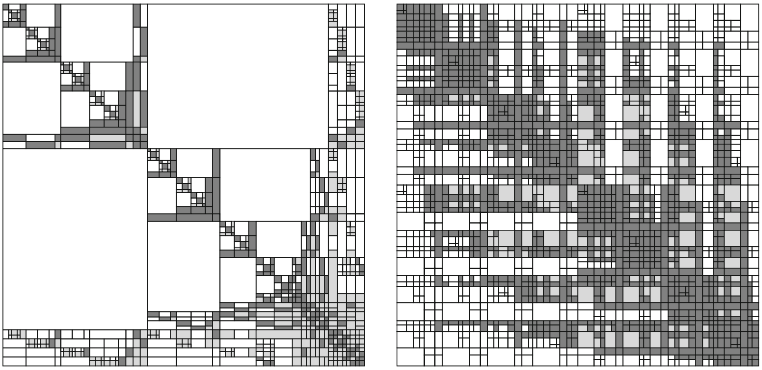


Fig. 8 $\mathcal{H}$-matrix L and U factors for matrices of size $N = 1{,}000$ ($n_{\min} = 20$) for domain decomposition based clustering (*left*) and for the standard bisection based clustering (*right*). *Dark grey blocks* are stored as full matrices whereas *light grey blocks* are stored in Rk-matrix format. *White blocks* are zero

Table 1 Comparison of geometric bisection and nested dissection for the Cholesky factorization with $\|I - A(LL^T)^{-1}\|_2 \sim 10^{-1}$

	N	Geometric bisection			Nested dissection		
		δ	Time (s)	Memory (MB)	δ	Time (s)	Memory (MB)
$d = 2$	253^2	$3_{10}{-}3$	3.1	70	$7_{10}{-}3$	1.0	50
	358^2	$1_{10}{-}3$	9.0	157	$3_{10}{-}3$	1.9	85
	511^2	$7_{10}{-}4$	20.4	349	$2_{10}{-}3$	4.7	212
	729^2	$3_{10}{-}4$	57.1	792	$1_{10}{-}3$	9.3	366
	$1{,}023^2$	$1_{10}{-}4$	135.2	1,730	$5_{10}{-}4$	21.0	873
	$1{,}447^2$	$1_{10}{-}4$	351.4	3,680	$2_{10}{-}4$	42.7	1,470
$d = 3$	40^3	$3_{10}{-}2$	29.2	188	$5_{10}{-}2$	9.1	119
	51^3	$1_{10}{-}2$	123.7	463	$3_{10}{-}2$	32.5	267
	64^3	$9_{10}{-}3$	310.2	1,080	$2_{10}{-}2$	79.0	503
	81^3	$5_{10}{-}3$	915.2	2,740	$1_{10}{-}2$	197.4	1,280
	102^3	$3_{10}{-}3$	2,797.5	6,280	$9_{10}{-}3$	481.8	2,670

In the second case, a direct solver is sought. For this, the Cholesky factorization is computed with a precision of the order of the discretization error. The latter is computed for the solution

$$u(x) := \begin{cases} x_1(1-x_1)x_2(1-x_2), & x \in \Omega =]0,1[^2, \\ x_1(1-x_1)x_2(1-x_2)x_3(1-x_3), & x \in \Omega =]0,1[^3. \end{cases}$$

The results for this computation can be found in Table 2. Due to memory constraints, some computations were not possible. These are marked with "n.c.".

The behavior of the $\mathcal{H}$-matrix arithmetic for both clustering methods is similar to the previous case, with the advantage of the nested dissection clustering being even more apparent.

To demonstrate the stability of the clustering based on nested dissection, a convection-diffusion problem

$$-\kappa \Delta u + b \cdot \nabla u = f \quad \text{in} \quad \Omega =]0,1[^d, \quad d \in \{2,3\}$$

is considered. For a fixed dimension ($n = 1023^2$ in $\mathbb{R}^2$ and $n = 63^3$ in $\mathbb{R}^3$), the $\mathcal{H}$-matrices are constructed and decomposed for a decreasing value of κ and hence an increasing dominance of the convection. The relative inversion error due to the $\mathcal{H}$-arithmetic is fixed to $\delta = 10^{-4}$, which lies between the previous two cases. Again, the standard admissibility with $\eta = 2$ is chosen for the construction of the block cluster tree.

Table 2 Comparison of geometric bisection and nested dissection for the Cholesky factorization with $\|I - A(LL^T)^{-1}\|_2 \sim \|u - u_h\|_2$

	N	Geometric bisection			Nested dissection		
		δ	Time (s)	Memory (MB)	δ	Time (s)	Memory (MB)
$d = 2$	253^2	$2_{10}{-7}$	7.4	99	$7_{10}{-7}$	1.3	54
	358^2	$5_{10}{-8}$	20.5	224	$1_{10}{-7}$	3.3	94
	511^2	$2_{10}{-8}$	48.6	513	$3_{10}{-8}$	7.0	229
	729^2	$4_{10}{-9}$	149.3	1,190	$4_{10}{-9}$	17.3	410
	$1{,}023^2$	$1_{10}{-9}$	322.3	2,600	$2_{10}{-9}$	33.6	948
	$1{,}447^2$	$6_{10}{-10}$	1,677.4	5,710	$9_{10}{-10}$	77.0	1,650
$d = 3$	40^3	$1_{10}{-5}$	184.0	436	$4_{10}{-5}$	39.3	171
	51^3	$5_{10}{-6}$	650.1	1,130	$1_{10}{-5}$	138.1	423
	64^3	$3_{10}{-6}$	1,950.8	2,760	$5_{10}{-6}$	422.4	908
	81^3	$1_{10}{-6}$	5,408.3	6,880	$2_{10}{-6}$	999.1	2,120
	102^3	n.c.	n.c.	n.c.	$9_{10}{-7}$	2,554.6	4,750

Two different types of convection are used in this example:

$$b_1(x) := \begin{pmatrix} 1 - x_2 \\ x_1 \end{pmatrix} \quad \text{and} \quad b_2(x) := \begin{pmatrix} 0.5 - x_2 \\ x_1 - 0.5 \end{pmatrix}.$$

A non-constant, non-circular convection is described by b_1, whereas b_2 represents a circular convection field. In $\mathbb{R}^3$, the third component of b_1 and b_2 is zero. Figure 9 shows the required times for the $\mathcal{H}$-LU factorizations.

As can be seen, the runtime in the case of a non-circular convection decreases with κ until an almost constant time is reached. For the circular convection, the behavior is slightly different. Here, the runtime first grows if κ is decreased. The maximal value is

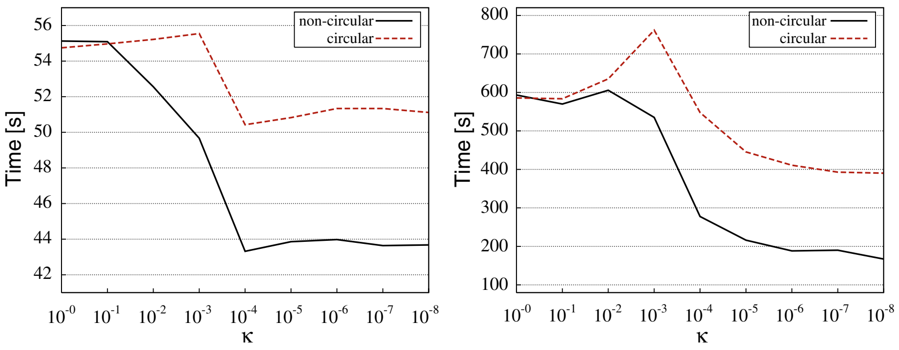


Fig. 9 Time for the $\mathcal{H}$-LU factorization for a convection-diffusion problem for an increasing convection domination in $\mathbb{R}^2$ (*left*) and $\mathbb{R}^3$ (*right*)

reached for $\kappa = 10^{-3}$. Afterwards, the runtime also decreases with the growing dominance of the convection term. We conclude that the time for the $\mathcal{H}$-LU factorization varies only little with a change of κ.

References

1. Bebendorf, M.: Why approximate LU decompositions of finite element discretizations of elliptic operators can be computed with almost linear complexity. SIAM J. Numer. Anal. **45**, 1472–1494 (2007)
2. Bebendorf, M., Hackbusch, W.: Existence of $\mathcal{H}$-matrix approximants to the inverse FE-matrix of elliptic operators with L^∞-coefficients. Numer. Math. **95**, 1–28 (2003)
3. Börm, S.: $\mathcal{H}^2$-matrix arithmetics in linear complexity. Computing **77**, 1–28 (2006)
4. Brainman, I., Toledo, S.: Nested-dissection orderings for sparse LU with partial pivoting. SIAM J. Math. Anal. Appl. **23**, 998–1012 (2002)
5. Gavrilyuk, I., Hackbusch, W., Khoromskij, B.: $\mathcal{H}$-matrix approximation for the operator exponential with applications. Numer. Math. **92**, 83–111 (2002)
6. George, A.: Nested dissection of a regular finite element mesh. SIAM J. Numer. Anal. **10**, 345–363 (1973)
7. Grasedyck, L.: Theorie und Anwendungen Hierarchischer Matrizen. Ph.D. thesis, Universität Kiel (2001)
8. Grasedyck, L., Le Borne, S.: $\mathcal{H}$-matrix preconditioners in convection-dominated problems. SIAM J. Math. Anal. **27**, 1172–1183 (2005)
9. Grasedyck, L., Hackbusch, W.: Construction and arithmetics of $\mathcal{H}$-matrices. Computing **70**, 295–334 (2003)
10. Grasedyck, L., Hackbusch, W., Kriemann, R.: Performance of $\mathcal{H}$-LU preconditioning for sparse matrices. Comput. Methods Appl. Math. **8**, 336–349 (2008)
11. Grasedyck, L., Kriemann, R., Le Borne, S.: Parallel black box $\mathcal{H}$-LU preconditioning for elliptic boundary value problems. Comput. Visual. Sci. **11**(4–6), 273–291 (2008)
12. Hackbusch, W.: A sparse matrix arithmetic based on $\mathcal{H}$-matrices. Part I: Introduction to $\mathcal{H}$-matrices. Computing **62**, 89–108 (1999)
13. Hackbusch, W.: Direct domain decomposition using the hierarchical matrix technique. In: Herrera, I., Keyes, D., Widlund, O., Yates, R. (eds.) Domain Decomposition Methods in Science and Engineering, UNAM, pp. 39–50 (2003)
14. Hackbusch, W., Khoromskij, B.: $\mathcal{H}$-matrix approximation on graded meshes. In: Whiteman, J.R. (ed.) The Mathematics of Finite Elements and Applications, pp. 307–316. Elsevier, Amsterdam (2000)
15. Hackbusch, W., Khoromskij, B.: A sparse matrix arithmetic based on $\mathcal{H}$-matrices. Part II: Application to multi-dimensional problems. Computing **64**, 21–47 (2000)
16. Hackbusch, W., Khoromskij, B., Sauter, S.: On $\mathcal{H}^2$-matrices. In: Bungartz, H., Hoppe, R., Zenger, C. (eds.) Lectures on Applied Mathematics, pp. 9–29. Springer, Berlin (2000)
17. Hendrickson, B., Rothberg, E.: Improving the run time and quality of nested dissection ordering. SIAM J. Sci. Comp. **20**, 468–489 (1998)
18. Ibragimov, I., Rjasanow, S., Straube, K.: Hierarchical Cholesky decomposition of sparse matrices arising from curl-curl-equations. J. Numer. Math. **15**, 31–58 (2007)
19. Le Borne, S.: Hierarchical matrices for convection-dominated problems. In: Kornhuber, R., Hoppe, R., Périaux, J., Pironneau, O., Widlund, O., Xu, J. (eds.) Domain Decomposition Methods in Science and Engineering. Lecture Notes in Computational Science and Engineering, pp. 631–638 (2004)
20. Le Borne, S., Grasedyck, L., Kriemann, R.: Domain-decomposition based H-LU preconditioners. In: Widlund, O.B., Keyes, D.E. (eds.) Domain Decomposition Methods in Science and Engineering XVI. Lecture Notes in Computational Science and Engineering, vol. 55, pp. 661–668. Springer, Berlin (2006)
21. Lipton, R.J., Rose, D.J., Tarjan, R.E.: Generalized nested dissection. SIAM J. Numer. Anal. **16**, 346–358 (1979)