Max-Planck-Institut
für Mathematik
in den Naturwissenschaften
Leipzig

Why approximate *LU* decompositions of finite element discretizations of elliptic operators can be computed with almost linear complexity

by

Mario Bebendorf

Preprint no.: 8 2005

Why approximate LU decompositions of finite element discretizations of elliptic operators can be computed with almost linear complexity*

Mario Bebendorf†

January 24, 2005

Abstract

Although the asymptotic complexity of direct methods for the solution of large sparse finite element systems arising from second-order elliptic partial differential operators is far from being optimal, these methods are often preferred over modern iterative methods. This is mainly due to their robustness. In this article it is shown that an (approximate) LU decomposition exists and that it can be computed in the algebra of hierarchical matrices with almost linear complexity and with the same robustness as the classical LU decomposition.

Mathematics Subject Classification (2000): 35C20, 65F05, 65F50, 65N30
Keywords: Approximate LU decomposition, fast direct solution, preconditioning, non-smooth coefficients, hierarchical matrices

1 Introduction

This article deals with the efficient solution of large sparse linear systems

$$Ax = b, \quad A \in \mathbb{R}^{n\times n}, \tag{1}$$

arising from the discretization of general second-order elliptic partial differential operators

$$Du = -\operatorname{div}\left[C\nabla u + c'u\right] + c''\cdot\nabla u + c_0 u \tag{2}$$

with coefficients $c_{ij}, c_i', c_j'', c_0 \in L^\infty(\Omega)$, $i,j = 1,\dots,d$, on a bounded Lipschitz domain $\Omega \subset \mathbb{R}^d$. For the solution of (1) two main classes of methods, *iterative* and *direct*, can be distinguished. The latter are based on factorizations of the sparse coefficient matrix A into easily invertible matrices. These methods are widely used due to their robustness. However, they suffer from so-called *fill-in*, i.e., compared with the sparsity of A considerably more entries of the factors will be non-zero. This usually happens to all entries within the bandwidth of the original matrix, which for Galerkin matrices of operators (2) scales like $n^{1-1/d}$ even if the bandwidth has been reduced for instance by the reverse Cuthill-McKee algorithm, by the minimum degree algorithm or by nested dissection. Hence, the fill-in will lead to a computational complexity of order $n^{3-2/d}$. Constants, however, are extremely small, making direct methods the methods of choice

*This work was supported by the DFG priority program SPP 1146 "Modellierung inkrementeller Umformverfahren"

†Fakultät für Mathematik und Informatik, Universität Leipzig, Augustusplatz 10/11, D-04109 Leipzig, Germany, bebendorf@math.uni-leipzig.de

if n is not too large. This situation persists if we are to solve large scale problems in two spatial dimensions. Here, recent multifrontal solvers (see [1] and the references therein) can be used.

For higher dimensions usually *iterative* methods such as Krylov subspace methods are more efficient, especially if an approximate solution of relatively low accuracy is sought. The advantage of these solution techniques is that the coefficient matrix enters the computation only through the matrix-vector product. On the other hand the convergence rate and hence the number of iterations usually depends on certain properties such as the condition number of the coefficient matrix. Since D is a second-order operator, the condition number of the finite element Galerkin matrix A grows like $n^{2/d}$ for large n, but also depends significantly on the coefficients of D. Hence, preconditioning is necessary in order to obtain reasonable convergence rates. This can be achieved for instance by *multigrid* [13] or the Bramble, Pasciak and Xu (BPX) preconditioner [8] if the problem class is restricted to stiffness matrices of elliptic operators with smooth coefficients, where special properties of the operator can be exploited. In the case of non-smooth coefficients, these methods might still work but suffer from poor convergence rates if the respective method is not adapted to the coefficients. In the last years much work has been done to develop robust multilevel methods. The *algebraic multigrid* (AMG) solvers [19] try to achieve this robustness by mostly heuristic strategies. *Domain decomposition methods* [22] are commonly used if the computational domain can be subdivided into a small number of parts on each of which the coefficients do not vary too much. In this case a problem can be decomposed into smaller ones for the purpose of parallel processing.

A class of preconditioners that are based on the ideas of the LU decomposition while avoiding fill-in are the so-called *incomplete LU factorizations* (ILU), see [20]. The ILU overcomes the problem of fill-in by setting entries in the factors L and U outside of the sparsity pattern of A to zero. Although the ILU can be applied to any sparse coefficient matrix provided the factorization does not break down, it is well suited for M- and diagonally dominant matrices. Generating an ILU is extremely fast and improves the convergence rate. However, it does usually not lead to a bounded number of iterations.

The aim of this article is to present a new approach that merges the advantages of direct and iterative methods. We will present an approximate LU decomposition which on one side inherits the robustness of the classical LU decomposition and which does not require a grid hierarchy while one the other side has logarithmic-linear complexity independently of the spatial dimension. Depending on the chosen accuracy, the approximate LU decomposition can be used as a direct solver or as a preconditioner in iterative schemes. Since fill-in will also occur during an approximate LU decomposition, we make use of the structure of hierarchical matrices, by which appropriate dense matrices can be treated with almost linear complexity. Consequently, the bandwidths of the factors L and U will not be an issue.

In the last years fast methods for the treatment of large dense matrices $M \in \mathbb{R}^{n\times n}$ have considerably spread. After the introduction of the *fast multipole method* [18], numerous methods based on low-rank approximations

$$M_{ts} \approx UV^T$$

of appropriate subblocks M_{ts} in the rows and columns $t, s \subset \{1, \ldots, n\}$ of M, where $U \in \mathbb{R}^{t\times k}$, $V \in \mathbb{R}^{s\times k}$ and k is small compared with $|t|$ and $|s|$, have been developed. The ideas of the fast multipole method originally aiming at an efficient approximate evaluation of matrix-vector products have recently been extended to a structure called *hierarchical matrices* ($\mathcal{H}$-matrices), see [12, 15]. Basically, these are matrices that are low-rank on each block of a certain partition stemming from a recursive subdivision of the set of matrix indices. In addition to the efficient matrix-vector multiplication (also with the transposed matrix) this structure provides approximate operations such as matrix addition, matrix-matrix multiplication and matrix inversion of

dense matrices with almost linear complexity. Furthermore, $\mathcal{H}$-matrices can be stored in an almost linear amount of units of memory. The structure of $\mathcal{H}$-matrices has originally been applied to integral equations, see [4, 5]. Recently [6, 2] it was shown that the inverse of finite element discretizations of operators of type (2) can be approximated by $\mathcal{H}$-matrices with a blockwise rank that depends logarithmically on both, the number of unknowns n and the accuracy ε. Interestingly, this approximation is very robust with respect to non-smooth coefficients. The presented approximate LU decomposition can be computed in significantly less time while keeping the same robustness with respect to the coefficients of D.

The structure of the following part of this article is as follows: In Section 2 a brief review of the structure of $\mathcal{H}$-matrices will be given. All results and notations from the field of $\mathcal{H}$-matrices necessary for this article will be presented. In particular we will describe how a hierarchical matrix partition is built from an arbitrary quasi-uniform discretization of Ω. In contrast to multigrid methods, where coarse grid nodes would have to be identified, we only have to cluster geometrically neighbored degrees of freedom.

Our aim is to accelerate the usual LU decomposition by employing $\mathcal{H}$-matrices as approximants to the factors L and U. It is by no means obvious that L and U allow for such an approximation. This question will be answered in Section 3. For this purpose we first prove that each Schur complement in A can be approximated by $\mathcal{H}$-matrices. This relies on the fact that the inverse of A has this property. It will be seen that the knowledge that each Schur complement of a matrix can be approximated will be enough to show that the factors L and U have approximants in the set of $\mathcal{H}$-matrices. In contrast to the inverse of a finite element Galerkin matrix, the LU decomposition of it has no analytic equivalent. It is thus surprising that the matrix partition that has proved useful for elliptic problems can also be used for the approximation of the factors L and U. The complexity estimates show the same dependence on the coefficients of operator (2) as the estimates for the $\mathcal{H}$-inverse. Hence, the asymptotic complexity and the robustness of the hierarchical inverse is inherited by the $\mathcal{H}$-LU decomposition.

In Section 4 this result is used when generating the approximate LU decomposition by an approximate block LU decomposition procedure using the $\mathcal{H}$-arithmetic. Once the matrix partition has been generated from the mesh information, by this procedure an approximate LU decomposition can be obtained from any Galerkin matrix in a purely algebraic way. Finally, in Section 5 numerical results for elliptic partial differential operators with non-smooth coefficients will confirm our analysis. It will be seen that the proposed approximate LU decomposition can be computed, stored and used during forward/backward substitution with almost linear complexity. Compared with the hierarchical inverse, the $\mathcal{H}$-LU decomposition can be computed in significantly less time. Furthermore, a problem-independent number of iterations of the conjugate gradients method can be achieved if a low-precision $\mathcal{H}$-LU decomposition is used as a preconditioner. Since the proposed preconditioner is explicit, it is particularly efficient if the same system with many right hand sides has to be solved.

2 Hierarchical matrices

This section gives a brief overview over the structure of $\mathcal{H}$-matrices originally introduced by Hackbusch et al. [12, 15]. We will describe the two principles on which the efficiency of $\mathcal{H}$-matrices is based. These are the hierarchical partitioning of the matrix into blocks and the blockwise restriction to low-rank matrices. These principles were also used in the mosaic-skeleton method [23].

In this article we will consider matrices $A \in \mathbb{R}^{n\times n}$ with entries

$$a_{ij} = a(\varphi_j, \varphi_i), \quad i,j = 1,\dots,n, \tag{3}$$

where a is a bilinear form and φ_i are basis functions with support $X_i := \operatorname{supp} \varphi_i$, $i \in I := \{1, \ldots, n\}$. For this article it is crucial that the basis functions φ_i are locally supported. Matrices of type (3) arise for instance from the Galerkin method, which is frequently used to discretize operators of type (2). If a arises from the variational formulation of differential operators, then A is a sparse matrix. If a, however, incorporates a non-local operator, then A will be fully populated in general.

In order to be able to approximate each block $b = t \times s$, $t, s \subset I$, of A by a matrix of low rank, i.e.,

$$A_b \approx UV^T, \quad U \in \mathbb{R}^{t \times k}, V \in \mathbb{R}^{s \times k},$$

where k is small compared with $|t|$ and $|s|$, b has to satisfy a certain condition which is caused by the operator hidden in a. In the field of elliptic partial differential operators D the corresponding Green function $G(x, y)$ has an algebraic singularity for $x = y$. Hence, the following condition on $b = t \times s$ has proved useful:

$$\min\{\operatorname{diam} X_t, \operatorname{diam} X_s\} < \eta \operatorname{dist}(X_t, X_s), \tag{4}$$

where $\eta > 0$ is a given real number, which typically is chosen from the interval $[0.5, 1.5]$. Blocks (t, s) satisfying (4) will be called admissible. The support X_t of a cluster t is the union of the supports of the basis functions corresponding to the indices in t:

$$X_t := \bigcup_{i \in t} X_i.$$

The *far-field* $\mathcal{F}(t)$ of $t \subset I$ is defined as

$$\mathcal{F}(t) := \{i \in I : \eta \operatorname{dist}(X_i, X_t) > \operatorname{diam} X_t\}$$

and by $\mathcal{N}_\eta(t) := I \setminus \mathcal{F}_\eta(t)$ we denote the *near-field* of t. As usual we denote

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

Hence, (4) is equivalent to the condition $s \subset \mathcal{F}(t)$ or $t \subset \mathcal{F}(s)$. Note that condition (4) implies

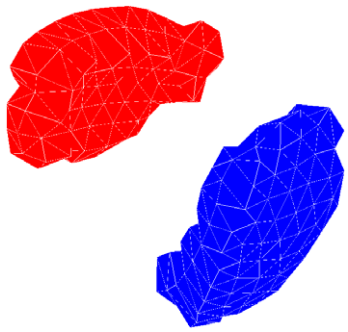


Figure 1: An admissible cluster pair (t, s)

that the partition we are looking for has to be refined towards the diagonal of A, since the diagonal entries arise from the interaction of the same basis functions, i.e., $\operatorname{dist}(X_t, X_s) = 0$ for all blocks $t \times s$ containing the diagonal.

2.1 The cluster tree

Since A cannot be approximated globally by a single low-rank matrix, we have to subdivide A into admissible blocks. One possibility is to recursively subdivide a block $b = t \times s$ into four subblocks $t_1 \times s_1$, $t_1 \times s_2$, $t_2 \times s_1$ and $t_2 \times s_2$, where $t = t_1 \cup t_2$ and $s = s_1 \cup s_2$, until its parts satisfy condition (4). Another possibility, which aims at generating blocks of largest possible size, is proposed in [5].

The rule how to subdivide a cluster t is given by the so called *cluster tree* T_I satisfying the following conditions:

(*i*) I is the root of T_I

(*ii*) if $t \in T_I$ is not a leaf, then t has sons $t_1, t_2 \in T_I$ so that $t = t_1 \cup t_2$ and $t_1 \cap t_2 = \emptyset$.

The set of sons of t is denoted by $\mathcal{S}(t)$, while $\mathcal{L}(T_I)$ stands for the set of leaves of the tree T_I. We assume contiguous clusters t, i.e., for $t \in T_I$ there is $t_{\min}, t_{\max} \in \mathbb{N}$ such that

$$t = \{i \in I : t_{\min} \leq i \leq t_{\max}\}. \tag{5}$$

A cluster tree is usually generated by recursive subdivision of I so as to minimize the diameter of each part. For practical purposes the recursion should be stopped if a certain cardinality $n_{\min}$ of the clusters is reached, rather than subdividing the clusters until only one index is left.

There a different methods for building the cluster tree. We favor the following strategy which is based on the *principal component analysis* and can be applied to arbitrary quasi-uniform sets X_i with centers $p_i \in \mathbb{R}^d$, $i \in I$. A cluster $t \subset I$ is subdivided by the hypersurface through its center

$$m_t := \frac{\sum_{i \in t} |X_i| p_i}{\sum_{i \in t} |X_i|}$$

with normal w_t, where w_t is the *main direction* of t, i.e., the vector maximizing

$$\sum_{i \in t} |w^T (p_i - m_t)|^2$$

with respect to w. Note that with $D_t := \sum_{i \in t} (p_i - m_t)(p_i - m_t)^T \in \mathbb{R}^{d \times d}$ it holds that

$$\sum_{i \in t} |w^T (p_i - m_t)|^2 = \sum_{i \in t} w^T (p_i - m_t)(p_i - m_t)^T w = w^T D_t w.$$

Hence, w_t is the eigenvector corresponding to the largest eigenvalue of the symmetric matrix D_t. Using w_t, the sons $\mathcal{S}(t) = \{t_1, t_2\}$ of t are defined by

$$t_1 = \{i \in t : w_t^T (p_i - m_t) > 0\}$$

and $t_2 := t \setminus t_1$. Note that the assumption (5) can be satisfied by moving the indices of t_1 to the beginning of t thereby reordering the index set I. The same division strategy is then recursively applied to the sons t_1, t_2 of t. The depth p of the resulting cluster tree T_I is of order $\log n$. The complexity of building it can be estimated as $\mathcal{O}(n \log n)$; cf. [3]. In the case of quasi-uniform meshes it can be seen that using this procedure the diameters of two clusters t and s from the same level of T_I are equivalent in the following sense: there is a constant $q \geq 1$ such that

$$\operatorname{diam} X_t \leq q \operatorname{diam} X_s. \tag{6}$$

Remark 2.1. *Since for each subdivision we have only two possibilities to arrange the indices, i.e, $t = [t_1, t_2]$ or $t = [t_2, t_1]$, the above construction leaves room for only $2^p n_{min}!$ permutations of I (the size of the leaves in T_I is assumed to be exactly $n_{\min}$). Hence, building the cluster tree determines the numbering of the indices in I up to $\mathcal{O}(n)$ permutations.*

2.2 The block cluster tree

Based on a cluster tree T_I which contains a hierarchy of partitions of I, we are able to construct the so called *block cluster tree* $T_{I\times I}$ describing a hierarchy of partitions of $I \times I$ by the following rule:

---

procedure build_block_cluster_tree($s \times t$)
begin
  if $t \times s$ is not admissible and $s, t \notin \mathcal{L}(T_I)$ **then**
    $\mathcal{S}(t \times s) := \{t' \times s' : t' \in \mathcal{S}(t),\ s' \in \mathcal{S}(s)\}$
    for $t' \times s' \in \mathcal{S}(t \times s)$ **do** build_block_cluster_tree($t' \times s'$)
  else $\mathcal{S}(t \times s) := \emptyset$
end

Applying *build_block_cluster_tree* to $I \times I$, we obtain a cluster tree for the index set $I \times I$. Upon completion of the algorithm, the set of leaves $P := \mathcal{L}(T_{I\times I})$ is a partition of $I \times I$ with blocks $b = t \times s \in P$ either satisfying (4) or consisting of clusters t and s with $\min\{|t|, |s|\} \leq n_{\min}$. For the generated blocks $b = t \times s \in P$ it holds that b is either on the diagonal ($t = s$), in the lower triangular part ($\min t \geq \max s$) or in the upper triangular part ($\max t \leq \min s$). The complexity of building the block cluster tree in the case of quasi-uniform grids can be estimated as $\mathcal{O}(\eta^{-d} n \log n)$; cf. [3].

We are now in a position to define the set of $\mathcal{H}$-matrices for a partition P with blockwise rank k

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

Note that $\mathcal{H}(P, k)$ is not a linear space since in general the sum of two rank-k matrices exceeds rank k.

Remark 2.2. *For a block $B \in \mathbb{R}^{t\times s}$ the low-rank representation $B = UV^T$, $U \in \mathbb{R}^{t\times k}$, $V \in \mathbb{R}^{s\times k}$, is only advantageous compared with the entrywise representation, if $k(|t| + |s|) \leq |t|\,|s|$. For the sake of simplicity in this article we will however assume that each block has the low-rank representation. Employing the entrywise representation for appropriate blocks will accelerate the algorithms.*

The cost of multiplying an $\mathcal{H}$-matrix $M \in \mathcal{H}(P, k)$ or its transposed M^T by a vector $x \in \mathbb{R}^n$ is inherited from the blockwise matrix-vector multiplication:

$$Mx = \sum_{t\times s\in P} M_{t\times s} x_s \quad \text{and} \quad M^T x = \sum_{t\times s\in P} (M_{t\times s})^T x_t.$$

Since each block $t\times s$ has the representation $M_{t\times s} = UV^T$, $U \in \mathbb{R}^{t\times k}$, $V \in \mathbb{R}^{s\times k}$ (see Remark 2.2), $\mathcal{O}(k(|t|+|s|))$ units of memory are needed to store $M_{t\times s}$. The matrix-vector multiplies $M_{t\times s}x_s = UV^T x_s$ and $(M_{t\times s})^T x_t = VU^T x_t$ can be done with $\mathcal{O}(k(|t| + |s|))$ operations. Exploiting the hierarchical structure of M, it can therefore be shown that both storing M and multiplying M or M^T by a vector has $\mathcal{O}(\eta^{-d} kn \log n)$ complexity. For a rigorous analysis the reader is referred to [3]. Therefore, $\mathcal{H}$-matrices are well suited for iterative schemes such as Krylov subspace methods.

2.3 Bandwidth and $\mathcal{H}$-matrices

Although $\mathcal{H}$-matrices are primarily aiming at dense matrices, the stiffness matrix A of the differential operator D from (2) is in $\mathcal{H}(P, n_{\min})$ and can therefore be stored in this format with complexity $\mathcal{O}(n)$. This can be seen by the following arguments. If $b \in P$ is admissible, then

the supports of the basis functions are pairwise disjoint. Hence, the matrix entries in this block vanish. In the remaining case, b does not satisfy (4). Then the size of one of the clusters is less or equal to $n_{\min}$. In either case, the rank of A_b does not exceed $n_{\min}$. The last observation is of particular importance since it will allow to compute an LU decomposition using approximate arithmetical operations on the set of $\mathcal{H}$-matrices, see Section 4.

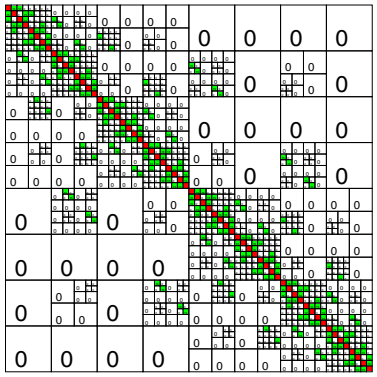


Figure 2: A sparse $\mathcal{H}$-matrix with its rank distribution

The efficiency of the usual LU decomposition is determined by the bandwidth of A. The reason for this is that although A is sparse, the factors L and U will in general be fully populated up to the bandwidth. Since $\mathcal{H}$-matrices are able to handle dense matrices with almost linear complexity, the bandwidth of A is not an issue when using this structure. Due to the reordering of indices required when building the cluster tree, we even obtain a bandwidth which is of order n. This will result in an enormous fill-in and is unavoidable as can be seen by the following example.

A matrix entry a_{ij} in the Galerkin matrix A will in general be non-zero if the supports of the associated basis functions φ_i and φ_j have a non-empty intersection. For simplicity we investigate the situation which occurs for a regular triangulation of the unit square in $\mathbb{R}^2$. Assume that after two subdivision steps this square has been subdivided into four smaller squares of the same size each containing $n/4$ supports. During the subdivision, the indices are reordered so that the k-th square contains the indices $(k-1)n/4+1$ to $kn/4$, $k = 1, \ldots, 4$. Hence, the first and the last square contain indices which differ by at least $n/2$. These squares intersect in the center of the original square. Therefore, at this point the supports of two basis functions φ_i and φ_j with $|i-j| \geq n/2$ intersect. This situation persists when the subdivision is continued since the indices are only rearranged within each subsquare.

2.4 Where can $\mathcal{H}$-matrices be applied ?

The structure of $\mathcal{H}$-matrices was originally designed to accelerate the building process and the matrix-vector multiplication of discrete integral operators with smooth kernels having an algebraic singularity at $x = y$. This kind of integral operator arises for instance from the boundary element method. For such operators the ACA algorithm [4, 5] can be used to generate the low-rank approximants from few of the original matrix entries.

In addition to discretizations of integral operators with smooth kernel functions, in [6, 2] it was shown that inverses of discrete elliptic differential operators with measurable coefficients can be approximated on partitions satisfying (4). Since the analysis of this article will be based on approximations of the inverse, we state the main result of [2]. Let the operator D from (2) be

a uniformly elliptic, i.e., for the coefficient $C(x) \in \mathbb{R}^{d\times d}$ of D it holds that C is symmetric with $c_{ij} \in L^\infty(\Omega)$ and

$$0 < \lambda \leq \lambda(x) \leq \Lambda$$

for all eigenvalues $\lambda(x)$ of $C(x)$ and almost all $x \in \Omega$.

Let $e_h(u) := \|u - P_h u\|_{L^2(\Omega)}$ be the finite element error, where $P_h : H_0^1(\Omega) \to V_h$ is the Ritz projector mapping $u \in H_0^1(\Omega)$ to its finite element solution, i.e., the solution of $a(u_h, v_h) = f(v_h)$ for all $v_h \in V_h$. We assume that the finite element method converges in the following sense

$$e_h(u) \leq \varepsilon_h \|f\|_{L^2(\Omega)} \quad \text{for all } u = L^{-1}f,\ f \in L^2(\Omega), \tag{7}$$

where $\varepsilon_h \to 0$ as $h \to 0$. Note that due to the lack of regularity of D we cannot assume a specific rate of convergence.

Theorem 2.3. *Let p be the depth of the cluster tree T_I defined in Section 2.1. Then there is a constant $c > 0$ defining $k := cp^2 \log^{d+1}(p/\varepsilon_h)$ and there is $C_{\mathcal{H}} \in \mathcal{H}(P, k)$ such that*

$$\|A^{-1} - C_{\mathcal{H}}\|_2 \leq c(D, \Omega, \eta)\, \varepsilon_h,$$

where $c(D, \Omega, \eta) > 0$ depends on the coefficients of D, the diameter of Ω and η. If $\varepsilon_h = \mathcal{O}(h^\beta)$ for some $\beta > 0$, $k = \mathcal{O}(\log^{d+3} n)$ holds.

In order to prove the previous theorem, the integral representation

$$D^{-1}\varphi(x) = \int_\Omega G(x, y)\varphi(y)\, \mathrm{d}y$$

with the Green function G of D and Ω was used. In contrast to operators arising from the boundary element method, the kernel function G is only locally in H^1 with respect to each variable. Nevertheless it is possible to prove that G and as a consequence the inverse of the finite element stiffness matrix can be approximated.

Remark 2.4. *Since the proof of Theorem 2.3 is based on the finite element error estimate* (7)*, we were only able to show existence of approximants with an accuracy which is of the order of the finite element error ε_h. This is not a restriction since a higher accuracy in the approximation of the inverse would be superposed by the finite element error in the solution anyway. However, numerical experiments show that the above result is true for any accuracy. Therefore, in this article we assume that for any $\varepsilon > 0$ there is $C_{\mathcal{H}} \in \mathcal{H}(P, k)$ with $k \sim |\log \varepsilon|^{d+1}(\log n)^2$ such that*

$$\|A^{-1} - C_{\mathcal{H}}\|_2 < c\varepsilon, \tag{8}$$

where $c > 0$ depends on the coefficients of D, the diameter of Ω and the cluster parameter η.

2.5 Schur complements

For domain decomposition methods (see for instance [22]), the efficient treatment of Schur complements is of particular importance. In this section it will be shown that Schur complements of subblocks of A can be approximated by $\mathcal{H}$-matrices. This result will lay ground to our main aim, the approximation of the factors L and U arising from the LU decomposition of A.

Assume that the Galerkin stiffness matrix $A \in \mathbb{R}^{n\times n}$ is partitioned in the following way:

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \tag{9}$$

where $A_{11} \in \mathbb{R}^{r\times r}$, $r \subset I$. We will show that the Schur complement

$$S = A_{22} - A_{21}A_{11}^{-1}A_{12}$$

can be approximated by an $\mathcal{H}$-matrix with blockwise rank k, where k depends only logarithmically on both, the approximation accuracy and n. For this purpose it is crucial to note that A_{11} in (9) is nothing but the Galerkin matrix of D if we replace Ω by the subdomain $X_r \times X_r$. Hence, Theorem 2.3 guarantees that an $\mathcal{H}$-matrix approximant for A_{11}^{-1} exists.

Let

$$N(t) = \{i \in I : \operatorname{dist}(X_i, X_t) = 0\}$$

denote a neighborhood of the cluster t. Since $\#t \geq n_{\min}$, for quasi-uniform meshes we may assume that

$$\operatorname{diam} X_{N(t)} \geq 3h, \tag{10}$$

where $h = \max_{i\in I} \operatorname{diam} X_i$. We need the following basic lemma, which states that the neighborhood of t is in the far-field of the neighborhood of s if t is in the far-field of s.

Lemma 2.5. *Let $0 < \eta < 1/(3+q)$, where q is defined in (6). If $t \subset \mathcal{F}_\eta(s)$, then*

$$N(t) \subset \mathcal{F}_{\tilde{\eta}}(N(s)), \quad \text{where} \quad \tilde{\eta} = \frac{3\eta}{1-(3+q)\eta}.$$

Proof. Let $x, y \in X_{N(s)}$. Since $\operatorname{dist}(x, X_s) \leq h$, we obtain

$$|x-y| \leq \operatorname{dist}(x, X_s) + \operatorname{dist}(y, X_s) + \operatorname{diam} X_s \leq \operatorname{diam} X_s + 2h.$$

Hence, $\operatorname{diam} X_{N(s)} \leq \operatorname{diam} X_s + 2h$. If $x \in X_{N(t)}$ and $y \in X_{N(s)}$, using (6) and (10) one has

$$\begin{aligned}
|x-y| &\geq \operatorname{dist}(X_t, X_s) - \operatorname{dist}(x, X_t) - \operatorname{diam} X_t - \operatorname{dist}(y, X_s) - \operatorname{diam} X_s \\
&\geq \frac{1}{\eta}(1-(1+q)\eta)\operatorname{diam} X_s - 2h \\
&\geq \frac{1}{\eta}(1-(1+q)\eta)(\operatorname{diam} X_{N(s)} - 2h) - 2h \\
&\geq \frac{1}{3\eta}(1-(3+q)\eta)\operatorname{diam} X_{N(s)},
\end{aligned}$$

which proves the assertion. □

By the following lemma (cf. [10]) it is possible to relate the blockwise spectral norm of an $\mathcal{H}$-matrix to its global norm.

Lemma 2.6. *If $\|A_b\|_2 \leq \varepsilon$ for all $b \in P$, then $\|A\|_2 \leq cp\varepsilon$, where p is the depth of the cluster tree T_I.*

Using the last two lemmas, we can now prove that the Schur complement S of finite element Galerkin matrices A can be approximated. For the proof we exploit the fact that the spectral norm of A is bounded with respect to n

$$\|A\|_2 \leq cn^{1-2/d}, \tag{11}$$

see for instance [14].

Theorem 2.7. *Let $A \in \mathbb{R}^{n\times n}$ be partitioned as in* (9)*. Then for the Schur complement*

$$S = A_{22} - A_{21}A_{11}^{-1}A_{12}$$

of A_{11} in A there is $S_{\mathcal{H}} \in \mathcal{H}(P,k)$, where $k \sim |\log\varepsilon|^{d+1}(\log n)^2$, such that

$$\|S - S_{\mathcal{H}}\|_2 < cpn^{2(1-2/d)}\varepsilon,$$

where $c > 0$ is independent of n and ε.

Proof. We have to show that for each admissible block $b = t \times s \in P$ in the rows and columns of A_{22} and any prescribed accuracy $\varepsilon > 0$ we can find a low-rank matrix which approximates S_b with accuracy ε. Since b is admissible, $(A_{22})_b = 0$ holds. Hence,

$$S_b = -A_{tr}A_{11}^{-1}A_{rs} = -\sum_{i,j\in r} A_{ti}(A_{11}^{-1})_{ij}A_{js}.$$

If $i \notin N(t)$, then $A_{ti} = 0$. If on the other hand $j \notin N(s)$, then $A_{js} = 0$. With the notation $N'(t) = N(t) \cap r$, we have

$$S_b = -\sum_{i\in N'(t), j\in N'(s)} A_{ti}(A_{11}^{-1})_{ij}A_{js}.$$

Since b is admissible, we have $t \subset \mathcal{F}_\eta(s)$ or $s \subset \mathcal{F}_\eta(t)$. According to Lemma 2.5, $N(t) \subset \mathcal{F}_{\tilde\eta}(N(s))$ or $N(s) \subset \mathcal{F}_{\tilde\eta}(N(t))$ holds. Following Theorem 2.3 (with η replaced by $\tilde\eta$), there is $U \in \mathbb{R}^{N'(t)\times k}$ and $V \in \mathbb{R}^{N'(s)\times k}$ with $k \sim |\log\varepsilon|^{d+1}(\log n)^2$ such that

$$\|(A_{11}^{-1})_{N'(t)N'(s)} - UV^T\|_2 < \varepsilon.$$

Let U and V be extended to $\hat U \in \mathbb{R}^{r\times k}$ and $\hat V \in \mathbb{R}^{r\times k}$ by adding zero rows. Observe that

$$A_{tr}\hat U\hat V^T A_{rs} = \sum_{i\in N'(t), j\in N'(s)} \sum_{\ell=1}^{k} A_{ti}U_{i\ell}V_{j\ell}A_{js} = \sum_{\ell=1}^{k} \left(\sum_{i\in N'(t)} A_{ti}U_{i\ell}\right)\left(\sum_{j\in N'(s)} V_{j\ell}A_{js}\right) = XY^T,$$

where $X \in \mathbb{R}^{t\times k}$, $Y \in \mathbb{R}^{s\times k}$ with the entries

$$X_{t\ell} = \sum_{i\in N'(t)} A_{ti}U_{i\ell} \quad \text{and} \quad Y_{s\ell} = \sum_{j\in N'(s)} V_{j\ell}A_{js}, \qquad \ell = 1,\dots,k.$$

Define $B \in \mathbb{R}^{r\times r}$ with entries

$$b_{ij} = \begin{cases} (A_{11}^{-1})_{ij}, & \text{if } i \in N'(t) \text{ and } j \in N'(s) \\ 0, & \text{else,} \end{cases}$$

then

$$\|S_b - XY^T\|_2 = \|A_{tr}(B - \hat U\hat V^T)A_{rs}\|_2 \le \|A_{tr}\|_2\|(A_{11}^{-1})_{N'(t)N'(s)} - UV^T\|_2\|A_{rs}\|_2 < \varepsilon\|A\|_2^2.$$

Using (11) and Lemma 2.6, we obtain the assertion. ∎

3 Hierarchical LU decomposition

Assume that all minors of A are non-zero. Then A can be decomposed as follows

$$A = LU,$$

where L is a lower-triangular and U is an upper-triangular matrix. In this subsection it will be shown that the factors L and U can be approximated by $\mathcal{H}$-matrices $L_{\mathcal{H}}$ and $U_{\mathcal{H}}$ if any Schur complement in A has this property. Note that the following proof consists of purely algebraic arguments only. Hence, the LU decompositions can be accelerated also for problems that do not stem from finite element applications as long as the Schur complement is known to have an approximant in the set of $\mathcal{H}$-matrices.

When computing pointwise LU decompositions, usually pivoting is performed in order to avoid zero or almost zero pivots. For block versions of the LU algorithm the possibilities of pivoting are limited if the blocking is given. In our case we can only choose from two possible pivots: block $t_1 \times t_1$ or block $t_2 \times t_2$ if the LU decomposition of a block $t \times t$, $t = t_1 \cup t_2$ is to be computed; cf. Remark 2.1. Hence, the accuracy analysis cannot rely on the advantages of pivoting.

In order to show that L and U can be approximated by $\mathcal{H}$-matrices it seems natural to define the approximants $L_{\mathcal{H}}$ and $U_{\mathcal{H}}$ recursively by replacing appropriate subblocks of the arising Schur complements with low-rank matrices. The error could then be estimated by the error analysis of the block LU decomposition, see [9]. The problem with this approach is that the arising Schur complements are not the original complements but complements that contain the perturbations from previous approximation steps. Since it cannot be guaranteed that these perturbed complements can be approximated by $\mathcal{H}$-matrices, we have to go a different way.

In the following subsection we first find a recursive relation between the Schur complement of a block b and the complements of its subblocks. An approximation of the Schur complement of b will then be defined by approximating the Schur complements on the leaves of b and using this recursive relation in order to bring the approximation to b. This means that we do not define an approximate LU decomposition by approximation in each step. Instead, we construct an exact LU decomposition of an appropriately perturbed original Galerkin matrix.

3.1 Approximating Schur complements hierarchically

Let $A \in \mathbb{R}^{n\times n}$ and $t, s \subset I$. With the notations $\hat{t} = \{i \in I : i \leq \max t\}$ and $\hat{s} = \{j \in I : j \leq \max s\}$ the Schur complement of the block $t \times s$ in $A_{\hat{t}\hat{s}}$ is defined as

$$S(t, s) = A_{ts} - A_{tr} A_{rr}^{-1} A_{rs}, \tag{12}$$

where $r = \{i \in I : i < \min\{t, s\}\}$, see Figure 3. Note that in the case $r = \emptyset$ this definition is meant to result in $S(t, s) = A_{ts}$.

Before we relate the Schur complement of a block to the complements of its subblocks, we state two estimates that will be important for the following error analysis. Let $A^{(k)}$ denote the matrix after k steps of the pointwise LU algorithm. For the stability of the LU decomposition the so-called growth factor (see for instance [16])

$$\rho_n := \max_{k=1,\dots,n-1} \frac{|A^{(k)}|}{|A|} \tag{13}$$

plays a central role. Here and in the following we use $|A| = \max_{i,j=1,\dots,n} |a_{ij}|$. As a consequence

$$|S(t, t)| \leq \rho_n |A| \quad \text{for all } t \subset I. \tag{14}$$

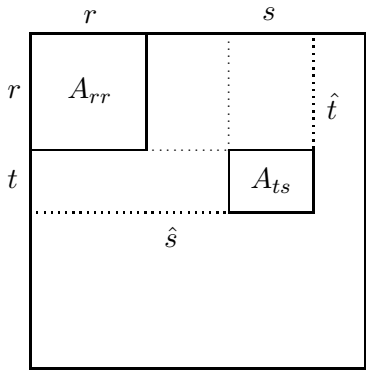


Figure 3: Schur complement of a block $t \times s$

A result due to Wilkinson states that $\rho_n \leq 2$ if A is diagonally dominant (by rows or by columns), see [16].

For $t \subset I$ define $\bar{t} = \{i \in I : i \geq \min t\}$ and $r = I \setminus \bar{t}$. Since A is assumed to be non-singular, each Schur complement in A is invertible. The $(2,2)$ block of the inverse of

$$A = \begin{bmatrix} A_{rr} & A_{r\bar{t}} \\ A_{\bar{t}r} & A_{\bar{t}\bar{t}} \end{bmatrix}$$

coincides with $S(\bar{t}, \bar{t})^{-1}$. Hence, we have

$$\|S(\bar{t}, \bar{t})^{-1}\|_2 \leq \|A^{-1}\|_2. \tag{15}$$

From definition (12) the following relation between the complement of a block and the complements of its subblocks can be obtained. For the ease of notation we first consider the case of blocks on the diagonal.

Lemma 3.1. *Let $t \in T_I$ and t_1, t_2 its sons. Then*

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

Proof. Let $S(t,t)$ be decomposed in the following way

$$S(t,t) = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix}.$$

It is obvious that $S_{11} = S(t_1,t_1)$, $S_{12} = S(t_1,t_2)$ and $S_{21} = S(t_2,t_1)$. Therefore, we only have to show that

$$S_{22} = S(t_2,t_2) + S_{21}S_{11}^{-1}S_{12}.$$

Let $r = \{i \in I : i < \min t\}$. Then from the definition of $S(t,t)$ it holds that

$$S(t_2,t_2) = A_{t_2t_2} - \begin{bmatrix} A_{t_2r} & A_{t_2t_1} \end{bmatrix} \begin{bmatrix} A_{rr} & A_{rt_1} \\ A_{t_1r} & A_{t_1t_1} \end{bmatrix}^{-1} \begin{bmatrix} A_{rt_2} \\ A_{t_1t_2} \end{bmatrix}.$$

Since

$$\begin{bmatrix} A_{rr} & A_{rt_1} \\ A_{t_1r} & A_{t_1t_1} \end{bmatrix}^{-1} = \begin{bmatrix} A_{rr}^{-1} & -A_{rr}^{-1}A_{rt_1}S_{11}^{-1} \\ 0 & S_{11}^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -A_{t_1r}A_{rr}^{-1} & I \end{bmatrix},$$

we have

$$\begin{aligned} S(t_2,t_2) &= A_{t_2t_2} - \begin{bmatrix} A_{t_2r} & A_{t_2t_1} \end{bmatrix} \begin{bmatrix} A_{rr}^{-1} & -A_{rr}^{-1}A_{rt_1}S_{11}^{-1} \\ 0 & S_{11}^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -A_{t_1r}A_{rr}^{-1} & I \end{bmatrix} \begin{bmatrix} A_{rt_2} \\ A_{t_1t_2} \end{bmatrix} \\ &= A_{t_2t_2} - \begin{bmatrix} A_{t_2r} & A_{t_2t_1} \end{bmatrix} \begin{bmatrix} A_{rr}^{-1} & -A_{rr}^{-1}A_{rt_1}S_{11}^{-1} \\ 0 & S_{11}^{-1} \end{bmatrix} \begin{bmatrix} A_{rt_2} \\ S_{12} \end{bmatrix} \\ &= A_{t_2t_2} - \begin{bmatrix} A_{t_2r}A_{rr}^{-1} & S_{21}S_{11}^{-1} \end{bmatrix} \begin{bmatrix} A_{rt_2} \\ S_{12} \end{bmatrix} = S_{22} - S_{21}S_{11}^{-1}S_{12}, \end{aligned}$$

which proves the assertion. □

Since a block $t \times s$ in the upper triangular part, i.e., $\max t \le \min s$, can be embedded into the block $r \times r$, $r = \{i \in I : \min t \le i \le \max s\}$, Lemma 3.1 gives

$$S(t,s) = \begin{bmatrix} S(t_1,s_1) & S(t_1,s_2) \\ S(t_2,s_1) + S(t_2,t_1)S(t_1,t_1)^{-1}S(t_1,s_1) & S(t_2,s_2) + S(t_2,t_1)S(t_1,t_1)^{-1}S(t_1,s_2) \end{bmatrix}.$$

Similarly, for a block $t \times s$ in the lower triangular part, i.e., $\max s \le \min t$, it holds that

$$S(t,s) = \begin{bmatrix} S(t_1,s_1) & S(t_1,s_2) + S(t_1,s_1)S(s_1,s_1)^{-1}S(s_1,s_2) \\ S(t_2,s_1) & S(t_2,s_2) + S(t_2,s_1)S(s_1,s_1)^{-1}S(s_1,s_2) \end{bmatrix}.$$

According to Theorem 2.7, for each admissible $b \in \mathcal{L}(T_{I\times I})$ the corresponding Schur complement $S(b)$ can be approximated by a matrix $\tilde{S}(b)$ of low rank, say k_S. If $b \in \mathcal{L}(T_{I\times I})$ does not satisfy (4), we set $\tilde{S}(b) = S(b)$. Following the recursive relation from Lemma 3.1, we define a hierarchy of approximants to the Schur complements in the following way:

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

for blocks $t \times t$ on the block diagonal. For blocks $t \times s$ in the upper triangular part we set accordingly

$$\tilde{S}(t,s) = \begin{bmatrix} \tilde{S}(t_1,s_1) & \tilde{S}(t_1,s_2) \\ \tilde{S}(t_2,s_1) + \tilde{S}(t_2,t_1)\tilde{S}(t_1,t_1)^{-1}\tilde{S}(t_1,s_1) & \tilde{S}(t_2,s_2) + \tilde{S}(t_2,t_1)\tilde{S}(t_1,t_1)^{-1}\tilde{S}(t_1,s_2) \end{bmatrix}, \tag{16}$$

and for blocks $t \times s$ in the lower part we define

$$\tilde{S}(t,s) = \begin{bmatrix} \tilde{S}(t_1,s_1) & \tilde{S}(t_1,s_2) + \tilde{S}(t_1,s_1)\tilde{S}(s_1,s_1)^{-1}\tilde{S}(s_1,s_2) \\ \tilde{S}(t_2,s_1) & \tilde{S}(t_2,s_2) + \tilde{S}(t_2,s_1)\tilde{S}(s_1,s_1)^{-1}\tilde{S}(s_1,s_2) \end{bmatrix}.$$

Note that $\tilde{S}(t_1,t_1)$ and $\tilde{S}(s_1,s_1)$ are invertible if ε is small enough.

In the following lemma we derive an estimate for $\|S(b) - \tilde{S}(b)\|_2$ in terms of the quantity

$$\kappa := \max\{\|S(t,t)^{-1}S(t,s)\|, \|S(s,t)S(t,t)^{-1}\|,\ t,s \in T_I \text{ satisfying } \max t \le \min s\}.$$

Lemma 3.2. *For all blocks $b \in T_{I\times I}$ it holds that*

$$\|S(b) - \tilde{S}(b)\|_2 \le 2^{\ell}(\kappa+1)^{2\ell}\varepsilon + O(\varepsilon^2), \tag{17}$$

where ℓ denotes the maximum distance of b to its leaves in the block cluster tree $T_{I\times I}$.

Proof. The assertion is proved by induction over ℓ. For $\ell = 0$, i.e., leaves in the block cluster tree, (17) is trivially satisfied, since $\tilde{S}(b)$ is just the low-rank approximant if b satisfies (4) or $\tilde{S}(b) = S(b)$ in the case that b is non-admissible.

Let b have height at most $\ell + 1$. Then the sons of b have height at most ℓ and satisfy the assertion due to the induction assumption. We consider blocks $b = t \times s$ on the diagonal, i.e., $t = s$. Then

$$S(b) = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} + S_{21}S_{11}^{-1}S_{12} \end{bmatrix} \quad \text{and} \quad \tilde{S}(b) = \begin{bmatrix} \tilde{S}_{11} & \tilde{S}_{12} \\ \tilde{S}_{21} & \tilde{S}_{22} + \tilde{S}_{21}\tilde{S}_{11}^{-1}\tilde{S}_{12} \end{bmatrix},$$

where for the ease of notation we set $S_{ij} = S(t_i, t_j)$ and $\tilde{S}_{ij} = \tilde{S}(t_i, t_j)$, $i, j = 1, 2$. As usual, t_1 and t_2 denote the sons of t. Due to the assumption, for the differences $E_{ij} := \tilde{S}_{ij} - S_{ij}$, $i, j = 1, 2$, it holds that

$$\|E_{ij}\|_2 \leq 2^\ell(\kappa + 1)^{2\ell}\varepsilon + O(\varepsilon^2).$$

With this notation

$$\tilde{S}(b) - S(b) = \begin{bmatrix} E_{11} & E_{12} \\ E_{21} & E_{22} + D \end{bmatrix}, \quad \text{where} \quad D := \tilde{S}_{21}\tilde{S}_{11}^{-1}\tilde{S}_{12} - S_{21}S_{11}^{-1}S_{12}.$$

It remains to find a bound for $\|D\|_2$. Since

$$\begin{aligned} D &= (S_{21} + E_{21})\tilde{S}_{11}^{-1}(S_{12} + E_{12}) - S_{21}S_{11}^{-1}S_{12} \\ &= S_{21}\tilde{S}_{11}^{-1}S_{12} - S_{21}S_{11}^{-1}S_{12} + E_{21}\tilde{S}_{11}^{-1}\tilde{S}_{12} + \tilde{S}_{21}\tilde{S}_{11}^{-1}E_{12} \\ &= S_{21}(\tilde{S}_{11}^{-1} - S_{11}^{-1})S_{12} + E_{21}\tilde{S}_{11}^{-1}S_{12} + S_{21}\tilde{S}_{11}^{-1}E_{12} + O(\varepsilon^2) \end{aligned}$$

and since $\tilde{S}_{11}^{-1}$ has the expansion

$$\tilde{S}_{11}^{-1} = S_{11}^{-1} - S_{11}^{-1}E_{11}S_{11}^{-1} + O(\varepsilon^2),$$

we obtain

$$\begin{aligned} \|D\|_2 &\leq 2^\ell(\kappa + 1)^{2\ell}\varepsilon \left(\|S_{21}S_{11}^{-1}\|_2\|S_{11}^{-1}S_{12}\|_2 + \|S_{11}^{-1}S_{12}\|_2 + \|S_{21}S_{11}^{-1}\|_2\right) + O(\varepsilon^2) \\ &\leq 2^\ell(\kappa + 1)^{2\ell}\varepsilon\kappa(\kappa + 2) + O(\varepsilon^2). \end{aligned}$$

This leads to

$$\begin{aligned} \|\tilde{S}(b) - S(b)\|_2 = \| \begin{bmatrix} E_{11} & E_{12} \\ E_{21} & E_{22} + D \end{bmatrix} \|_2 &\leq 2\max\{\|E_{11}\|_2, \|E_{12}\|_2, \|E_{21}\|_2, \|E_{22}\|_2 + \|D\|_2\} \\ &\leq 2\varepsilon 2^\ell(\kappa + 1)^{2\ell}\left(1 + \kappa(\kappa + 2)\right) + O(\varepsilon^2) = 2^{\ell+1}(\kappa + 1)^{2(\ell+1)}\varepsilon + O(\varepsilon^2). \end{aligned}$$

For blocks $b = t \times s$ in the upper or lower triangular part similar arguments apply. □

Let $b = t \times s$ be a block in the upper triangular part, i.e., $\max t \leq \min s$. Let $\hat{s} = \{i \in I : i > \max t\} \supset s$ and $\bar{t} = t \cup \hat{s} = \{i \in I : i \geq \min t\}$. The $(1, 2)$-block of the block inverse of

$$S(\bar{t}, \bar{t}) = \begin{bmatrix} S(t, t) & S(t, \hat{s}) \\ S(\hat{s}, t) & S(\hat{s}, \hat{s}) + S(\hat{s}, t)S(t, t)^{-1}S(t, \hat{s}) \end{bmatrix}$$

is $S(t, t)^{-1}S(t, \hat{s})S(\hat{s}, \hat{s})^{-1}$. Hence, using (14) and (15) we obtain

$$\begin{aligned} \|S(t, t)^{-1}S(t, s)\|_2 &\leq \|S(t, t)^{-1}S(t, \hat{s})\|_2 = \|(S(\bar{t}, \bar{t})^{-1})_{12}S(\hat{s}, \hat{s})\|_2 \\ &\leq \|S(\bar{t}, \bar{t})^{-1}\|_2\|S(\hat{s}, \hat{s})\|_2 \leq n\rho_n\|A^{-1}\|_2\|A\|_2 = n\rho_n\mathrm{cond}_2(A). \end{aligned}$$

For blocks $b = t \times s$ in the lower triangular part the same arguments apply. Hence, we obtain the following worst-case estimate for κ

$$\kappa \leq n\rho_n \mathrm{cond}_2(A).$$

If A is symmetric positive definite, then each Schur complement $S(t,t)$, $t \subset I$ is symmetric positive definite. With the same arguments as above one has

$$\begin{aligned}
\|S(t,t)^{-1}S(t,s)\|_2 &\leq \|(S(\bar{t},\bar{t})^{-1})_{12}S(\hat{s},\hat{s})\|_2 = \|(S(\bar{t},\bar{t})^{-1/2}S(\bar{t},\bar{t})^{-1/2})_{12}S(\hat{s},\hat{s})\|_2 \\
&\leq \|S(\bar{t},\bar{t})^{-1/2}\|_2 \| \begin{bmatrix} (S(\bar{t},\bar{t})^{-1/2})_{1,2} \\ (S(\bar{t},\bar{t})^{-1/2})_{2,2} \end{bmatrix} S(\hat{s},\hat{s})\|_2 \\
&= \|S(\bar{t},\bar{t})^{-1}\|_2^{1/2} \|S(\hat{s},\hat{s})(S(\bar{t},\bar{t})^{-1})_{22}S(\hat{s},\hat{s})\|_2^{1/2} \\
&= \|S(\bar{t},\bar{t})^{-1}\|_2^{1/2} \|S(\hat{s},\hat{s})\|_2^{1/2}.
\end{aligned}$$

The last equality follows from the fact that $(S(\bar{t},\bar{t})^{-1})_{22} = S(\hat{s},\hat{s})^{-1}$. For symmetric positive definite A one has $\|S(\hat{s},\hat{s})\|_2 \leq \|A_{\hat{s}\hat{s}}\|_2 \leq \|A\|_2$. Hence, together with (15) we obtain

$$\kappa \leq \sqrt{\mathrm{cond}_2(A)}.$$

3.2 Constructing the factors $L_{\mathcal{H}}$ and $U_{\mathcal{H}}$

Based on the approximate Schur complements $\tilde{S}(b)$, $b \in T_{I\times I}$, we construct the factors $L_{\mathcal{H}}$ and $U_{\mathcal{H}}$ of the LU decomposition of $\tilde{A} := \tilde{S}(I,I)$ by the following recursion. In order to define the factors $\tilde{L}$ and $\tilde{U}$ of $\tilde{S}(t,t) = \tilde{L}\tilde{U}$, $t \in T_I \setminus \mathcal{L}(T_I)$, we set

$$\tilde{L} := \begin{bmatrix} \tilde{L}_1 & 0 \\ \tilde{S}(t_2,t_1)\tilde{U}_1^{-1} & \tilde{L}_2 \end{bmatrix} \quad \text{and} \quad \tilde{U} := \begin{bmatrix} \tilde{U}_1 & \tilde{L}_1^{-1}\tilde{S}(t_1,t_2) \\ 0 & \tilde{U}_2 \end{bmatrix}, \tag{18}$$

where

$$\tilde{L}_1\tilde{U}_1 = \tilde{S}(t_1,t_1), \qquad \tilde{L}_2\tilde{U}_2 = \tilde{S}(t_2,t_2)$$

and t_1, t_2 are the sons of t. If $t \in \mathcal{L}(T_I)$ then $\tilde{L}$ and $\tilde{U}$ are defined by the pointwise LU decomposition. Note that since

$$\begin{bmatrix} \tilde{L}_1 & 0 \\ \tilde{S}(t_2,t_1)\tilde{U}_1^{-1} & \tilde{L}_2 \end{bmatrix} \begin{bmatrix} \tilde{U}_1 & \tilde{L}_1^{-1}\tilde{S}(t_1,t_2) \\ 0 & \tilde{U}_2 \end{bmatrix} = \begin{bmatrix} \tilde{L}_{11}\tilde{U}_{11} & \tilde{S}(t_1,t_2) \\ \tilde{S}(t_2,t_1) & \tilde{L}_{22}\tilde{U}_{22} + \tilde{S}(t_2,t_1)\tilde{S}(t_1,t_1)^{-1}\tilde{S}(t_1,t_2) \end{bmatrix},$$

due to Lemma 3.1 we obtain $\tilde{L}\tilde{U} = \tilde{S}(t,t)$.

The following lemma shows that the off-diagonal blocks in (18) are blockwise low-rank. Since the diagonal blocks $\tilde{L}_1, \tilde{L}_2$ and $\tilde{U}_1, \tilde{U}_2$ have the same structure as $\tilde{L}$ and $\tilde{U}$, respectively, it follows that $\tilde{L}$ and $\tilde{U}$ are $\mathcal{H}$-matrices.

Lemma 3.3. *Let X, Y solve $\tilde{L}X = \tilde{S}(t,s)$ and $Y\tilde{U} = \tilde{S}(s,t)$, where $\max t \leq \min s$. Then X and Y are $\mathcal{H}$-matrices with blockwise rank at most k_S, where k_S is the rank that was used in the construction of $\tilde{S}$.*

Proof. We prove the assertion by induction. If $t \times s$ is a leaf, then $\tilde{S}(t,s)$ is a matrix of rank at most k_S. Hence, the rank of $X = \tilde{L}^{-1}\tilde{S}(t,s)$ cannot exceed k_S. If $t \times s$ is not a leaf, t has sons t_1 and t_2. If we define $X_1 \in \mathbb{R}^{t_1\times s}$ and $X_2 \in \mathbb{R}^{t_2\times s}$ by

$$\tilde{L}_1X_1 = \tilde{S}(t_1,s) \quad \text{and} \quad \tilde{L}_2X_2 = \tilde{S}(t_2,s),$$

respectively, we know by induction that X_1 and X_2 are $\mathcal{H}$-matrices. Hence,

$$X := \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$$

is an $\mathcal{H}$-matrix satisfying

$$\tilde{L}X = \begin{bmatrix} \tilde{L}_1 & 0 \\ \tilde{S}(t_2,t_1)\tilde{U}_1^{-1} & \tilde{L}_2 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} \tilde{S}(t_1,s) \\ \tilde{S}(t_2,s) + \tilde{S}(t_2,t_1)\tilde{S}(t_1,t_1)^{-1}\tilde{S}(t_1,s) \end{bmatrix} = \tilde{S}(t,s)$$

due to the definition (18) of $\tilde{L}$ and (16). The proof for Y can be done analogously. □

The following theorem is the main result of this article.

Theorem 3.4. *Assume that any Schur complement $S(b)$, $b \in P$ admissible, of A can be approximated by a matrix of rank k with accuracy ε such that $k \sim (\log n)^{\alpha} |\log \varepsilon|^{\beta}$, $\alpha, \beta > 0$. Then there are lower and upper triangular matrices $L_{\mathcal{H}}, U_{\mathcal{H}} \in \mathcal{H}(P, k')$ with*

$$k' \sim (\log n)^{\alpha+\beta} |\log \varepsilon|^{\beta} (\log(n\rho_n \mathrm{cond}_2(A)))^{\beta}$$

such that

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

Proof. According to Lemma 3.3 there are $\mathcal{H}$-matrices $L_{\mathcal{H}}, U_{\mathcal{H}} \in \mathcal{H}(P,k)$ satisfying $\tilde{S}(I,I) = L_{\mathcal{H}} U_{\mathcal{H}}$. Since $A = S(I,I)$, we have

$$\|A - L_{\mathcal{H}} U_{\mathcal{H}}\| = \|S(I,I) - \tilde{S}(I,I)\| \leq 2^p (n\rho_n \mathrm{cond}_2(A) + 1)^{2p} \varepsilon, \tag{19}$$

where $p \sim \log n$ is the depth of the cluster tree. □

Remark 3.5. *We have seen in Theorem 2.7 that in the case of finite element discretizations of elliptic partial differential operators of type* (2) *each Schur complement S in A can be approximated by an $\mathcal{H}$-matrix. Hence, the last theorem can be applied to such matrices. Compared with the rank of the inverse the blockwise rank of the factors $L_{\mathcal{H}}$ and $U_{\mathcal{H}}$ bears an additional factor $(\log(n))^{2(d+1)}$ provided ρ_n is bounded. However, from the numerical experiments it will be seen that the complexity of the $\mathcal{H}$-LU decomposition in practice is much smaller than the complexity of the $\mathcal{H}$-inverse.*

4 Computing the hierarchical LU decomposition

In the last section we have seen that the factor L and U from an LU decomposition of A can be approximated by $\mathcal{H}$-matrices $L_{\mathcal{H}}$ and $U_{\mathcal{H}}$ whenever the Schur complements in A possess this property. Although the construction used for the proof could in principle be used to compute $L_{\mathcal{H}}$ an $U_{\mathcal{H}}$, for an improved efficiency we rather use another method which is based on the block LU decomposition, i.e., on the recursion (18).

On the set $\mathcal{H}(P,k)$ of hierarchical matrices approximate versions of the usual matrix operations like addition, matrix-matrix multiplication and inversion can be defined; cf. [12, 15, 11]. The truncation precision these operations are performed with will be denoted by $\varepsilon_{\mathcal{H}}$. The hierarchical LU decomposition can then be computed using these operations during the block LU decomposition instead of the usual ones.

In order to define the $\mathcal{H}$-*LU* decomposition we exploit the hierarchical block structure of a block A_{tt}, $t \in T_I \setminus \mathcal{L}(T_I)$:

$$A_{tt} = \begin{bmatrix} A_{t_1t_1} & A_{t_1t_2} \\ A_{t_2t_1} & A_{t_2t_2} \end{bmatrix} = \begin{bmatrix} L_{t_1t_1} & \\ L_{t_2t_1} & L_{t_2t_2} \end{bmatrix} \begin{bmatrix} U_{t_1t_1} & U_{t_1t_2} \\ & U_{t_2t_2} \end{bmatrix},$$

where $t_1, t_2 \in T_I$ denote the sons of t in T_I. Hence, the *LU* decomposition of a block A_{tt} is reduced to the following four problems on the sons of $t \times t$:

(*i*) Compute $L_{t_1t_1}$ and $U_{t_1t_1}$ from the *LU* decomposition $L_{t_1t_1}U_{t_1t_1} = A_{t_1t_1}$.

(*ii*) Compute $U_{t_1t_2}$ from $L_{t_1t_1}U_{t_1t_2} = A_{t_1t_2}$.

(*iii*) Compute $L_{t_2t_1}$ from $L_{t_2t_1}U_{t_1t_1} = A_{t_2t_1}$.

(*iv*) Compute $L_{t_2t_2}$ and $U_{t_2t_2}$ from the *LU* decomposition $L_{t_2t_2}U_{t_2t_2} = A_{t_2t_2} - L_{t_2t_1}U_{t_1t_2}$.

If a block $t \times t \in \mathcal{L}(T_{I\times I})$ is a leaf, the usual pivoted *LU* decomposition is employed. For (*i*) and (*iv*) two *LU* decompositions of half the size have to be computed. In oder to solve (*ii*), i.e., solve a problem of the structure $L_{tt}B_{ts} = A_{ts}$ for B_{ts}, where L_{tt} is a lower triangular matrix and $t \times s \in T_{I\times I}$, we use a recursive block forward substitution: If the block $t \times s$ is not a leaf in $T_{I\times I}$, from the decompositions of the blocks A_{ts}, B_{ts} and L_{tt} into their subblocks (t_1, t_2 and s_1, s_2 are again the sons of t and s, respectively)

$$\begin{bmatrix} L_{t_1t_1} & \\ L_{t_2t_1} & L_{t_2t_2} \end{bmatrix} \begin{bmatrix} B_{t_1s_1} & B_{t_1s_2} \\ B_{t_2s_1} & B_{t_2s_2} \end{bmatrix} = \begin{bmatrix} A_{t_1s_1} & A_{t_1s_2} \\ A_{t_2s_1} & A_{t_2s_2} \end{bmatrix}$$

one observes that B_{ts} can be found from the following equations

$$\begin{aligned} L_{t_1t_1}B_{t_1s_1} &= A_{t_1s_1} \\ L_{t_1t_1}B_{t_1s_2} &= A_{t_1s_2} \\ L_{t_2t_2}B_{t_2s_1} &= A_{t_2s_1} - L_{t_2t_1}B_{t_1s_1} \\ L_{t_2t_2}B_{t_2s_2} &= A_{t_2s_2} - L_{t_2t_1}B_{t_1s_2}, \end{aligned}$$

which are again of type (*ii*). If on the other hand $t \times s$ is a leaf, the usual forward substitution is applied. Similarly, one can solve (*iii*) by recursive block backward substitution.

The complexity of the above recursions is mainly determined by the complexity of the hierarchical matrix-matrix multiplication, which can be estimated as $\mathcal{O}(k^2 n \log^2 n)$ for two matrices from $\mathcal{H}(P,k)$; cf. [11]. Each operation is carried out with precision $\varepsilon_{\mathcal{H}}$. A result [9] on the stability analysis of the block *LU* decomposition states that the product *LU* is backward stable in the following sense

$$\|A - LU\|_2 < c(n)\varepsilon_{\mathcal{H}}(\|A\|_2 + \|L\|_2\|U\|_2).$$

Provided $\|L\|_2\|U\|_2 \approx \|A\|_2$, the accuracy of *LU* will hence be of order $\varepsilon_{\mathcal{H}}$. Employing the $\mathcal{H}$-matrix arithmetic, it is therefore possible to generate an approximate *LU* decomposition of an $\mathcal{H}$-matrix $A \in \mathcal{H}(P,k)$ to any prescribed accuracy ε with almost linear complexity. The *LU* decomposition of $\mathcal{H}$-matrices of a format that is too restrictive for our needs has already been used in [17].

Remark 4.1. *Although the intermediate results of the $\mathcal{H}$-LU decomposition are guaranteed to be $\mathcal{H}$-matrices, the blockwise rank k is not known. Note that our theory cannot be applied to this construction of the LU decomposition since the computed Schur complements in (iv) are approximate ones. Nevertheless, it will be seen from the numerical experiments that k still depends logarithmically on both, the accuracy ε and the number of unknowns n.*

In the case of positive definite matrices A it is possible to define an $\mathcal{H}$-version of the Cholesky decomposition of a block A_{tt}, $t \in T_I \setminus \mathcal{L}(T_I)$:

$$A_{tt} = \begin{bmatrix} A_{t_1t_1} & A_{t_1t_2} \\ A_{t_1t_2}^T & A_{t_2t_2} \end{bmatrix} = \begin{bmatrix} L_{t_1t_1} & \\ L_{t_2t_1} & L_{t_2t_2} \end{bmatrix} \begin{bmatrix} L_{t_1t_1} & \\ L_{t_2t_1} & L_{t_2t_2} \end{bmatrix}^T.$$

This factorization is recursively computed by

$$\begin{aligned} L_{t_1t_1} L_{t_1t_1}^T &= A_{t_1t_1} \\ L_{t_1t_1} L_{t_2t_1}^T &= A_{t_1t_2} \\ L_{t_2t_2} L_{t_2t_2}^T &= A_{t_2t_2} - L_{t_2t_1} L_{t_2t_1}^T \end{aligned}$$

using the usual Cholesky decomposition on the leaves of $T_{I\times I}$. The second equation $L_{t_1t_1} L_{t_2t_1}^T = A_{t_1t_2}$ is solved for $L_{t_2t_1}$ in a similar way as $U_{t_1t_2}$ has previously been obtained in the LU decomposition.

Once A has been decomposed, the solution of $Ax = b$ can be found by forward/backward substitution: $L_{\mathcal{H}} y = b$ and $U_{\mathcal{H}} x = y$. Since $L_{\mathcal{H}}$ and $U_{\mathcal{H}}$ are $\mathcal{H}$-matrices, y_t, $t \in T_I \setminus \mathcal{L}(T_I)$, can be computed recursively by solving the following systems for y_{t_1} and y_{t_2}

$$L_{t_1t_1} y_{t_1} = b_{t_1} \quad \text{and} \quad L_{t_2t_2} y_{t_2} = b_{t_2} - L_{t_2t_1} y_{t_1}.$$

If $t \in \mathcal{L}(T_I)$ is a leaf, a usual triangular solver is used. The backward substitution can be done analogously. The complexity of this forward/backward substitution is determined by the complexity of the hierarchical matrix-vector multiplication, which is $\mathcal{O}(kn \log n)$ if an $\mathcal{H}(P,k)$-matrix is multiplied by a vector.

4.1 Approximate direct or preconditioned iterative solution

The $\mathcal{H}$-LU decomposition can be used for preconditioning iterative schemes. Let $C = L_{\mathcal{H}} U_{\mathcal{H}}$, where $L_{\mathcal{H}}$ and $U_{\mathcal{H}}$ are lower and upper triangular $\mathcal{H}$-matrices such that $L_{\mathcal{H}} U_{\mathcal{H}} \approx A$ is an approximate LU decomposition. If A is symmetric positive definite, $C = L_{\mathcal{H}} L_{\mathcal{H}}^T$ is used as a preconditioner, where $L_{\mathcal{H}}$ is the lower triangular $\mathcal{H}$-matrix from the approximate Cholesky decomposition $L_{\mathcal{H}} L_{\mathcal{H}}^T \approx A$. Hence, during any Krylov subspace method like GMRES, in addition to multiplications of A and A^T by a vector, forward/backward substitutions have to be applied when applying $C^{-1} = (L_{\mathcal{H}} U_{\mathcal{H}})^{-1} = U_{\mathcal{H}}^{-1} L_{\mathcal{H}}^{-1}$.

Note that in order to generate a preconditioner it is not necessary to compute the $\mathcal{H}$-LU decomposition with high precision. Assume that we have computed a matrix C such that

$$\|I_n - AC^{-1}\|_2 \leq \delta < 1. \tag{20}$$

It can be shown, cf. [7], that

$$|\lambda_i(AC^{-1})| \geq 1 - \delta, \quad i = 1, \ldots, n, \tag{21}$$

where $\lambda_i(AC^{-1})$ denotes the i-th eigenvalue of AC^{-1}, and for the positive definite case

$$\operatorname{cond}_2(AC^{-1}) \leq \frac{1+\delta}{1-\delta}. \tag{22}$$

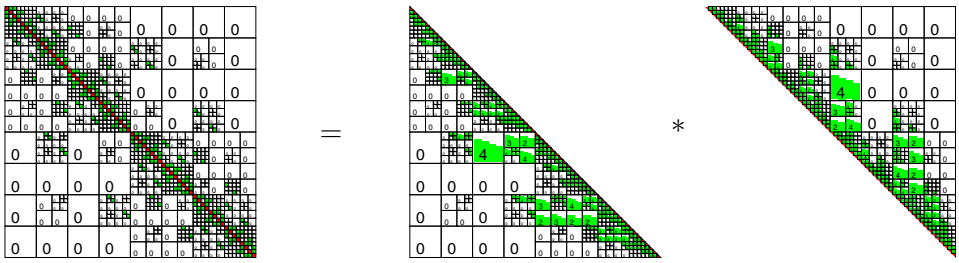
Hence, in order to obtain a spectrally equivalent preconditioner with almost linear complexity, the accuracy δ in (20) can be chosen independently of n, say $\delta = 0.1$. Note that the condition number will even be problem-independent, i.e., it will not only be bounded with respect to n

but also with respect to the coefficients of the operator in (2) and the computational domain Ω. If A is symmetric positive definite, C also has to be symmetric positive definite in order to be able to apply the conjugates gradient method. It can be shown [7] that C from (20) possess this property if for δ from (22) it holds that $\delta < 1/2$. Although in [7] preconditioners based on the hierarchical inverse were investigated, the same results also hold for preconditioners employing the hierarchical LU decomposition.

Since both, generating the $\mathcal{H}$-LU decomposition and the forward/backward substitution are efficient operations, one can equally use the $\mathcal{H}$-LU decomposition as a direct solver. In this case the accuracy ε which the $\mathcal{H}$-LU decomposition is generated with has to be of the same order as the finite element error ε_h. Once the factors have been computed, only forward/backward substitutions have to be applied. Hence, if $Ax = b$ is to be solved for many right-hand sides, the overall complexity of this approach might be less than an iterative solution using the $\mathcal{H}$-LU preconditioner.

5 Numerical results

In this section we make use of the hierarchical LU decompositions for preconditioning finite element stiffness matrices in two and three spatial dimensions. The emphasis in these tests is laid on robustness with respect to varying coefficients of the underlying operator.



All computations were carried out on an Athlon64 PC (2 GHz) with 4 GB of core memory. For compiling the $\mathcal{H}$-matrix library[1], the Intel compiler was used.

5.1 Two-dimensional diffusion

As a first set of tests we consider the Dirichlet boundary value problem

$$\begin{aligned} -\operatorname{div} \alpha(x) \nabla u &= 0 \quad \text{in } \Omega, \\ u &= f \quad \text{on } \partial\Omega, \end{aligned}$$

where $\Omega = (0,1)^2$ is the unit square in $\mathbb{R}^2$ and $\alpha(x)$ is a random number from the interval $[0, a]$ for $x = (x_1, x_2)$ satisfying $x_1 > x_2$. In the remaining part of Ω the coefficient α is set to 1. The amplitude a will be used to demonstrate that the presented method is not sensitive with respect to non-smooth coefficients.

The main aim of these two-dimensional tests is to show that the computational complexity of the presented hierarchical LU decomposition is almost linear, thereby confirming our estimates. In the following table we compare for different problem sizes n and for different amplitudes a the computational effort if the hierarchical LU decomposition is used for preconditioning the problem from above. Since the discrete operator is symmetric positive definite, we actually compute the

[1] A C++ implementation of the $\mathcal{H}$-matrix structure can be obtained from the following web-site `http://www.mathematik.uni-leipzig.de/~bebendorf/AHMED.html`

Cholesky decomposition LL^T. Table 1 shows the time (T_{LL^T}) for computing the hierarchical Cholesky decomposition, its memory consumption (MB) and the number of iterations of the conjugate gradients method (CG) required to reach a relative residual below $1e-4$. The minimal cluster size, see Subsection 2.1, was chosen to be $n_{\min} = 50$.

		$a = 1$				$a = 10^9$			
n	$\varepsilon_{\mathcal{H}}$	T_{LL^T}	MB	Its	T_{It}	T_{LL^T}	MB	Its	T_{It}
39 061	$7e-2$	0.9s	27.6	14	0.5s	0.9s	27.6	24	0.9s
78 961	$6e-2$	2.1s	56.6	19	1.5s	2.1s	57.1	31	2.4s
159 201	$5e-2$	5.6s	127.4	26	4.5s	5.6s	127.8	45	7.7s
318 096	$4e-2$	13.4s	267.9	19	9.0s	13.0s	267.1	36	14.1s
638 401	$3e-2$	32.1s	574.8	20	16.3s	33.8s	573.8	37	37.4s
1 276 900	$2e-2$	77.3s	1221.1	24	39.2s	84.2s	1216.3	45	90.6s
2 556 801	$1e-2$	230.3s	2774.5	28	115.6s	235.1s	2769.5	49	217.5s

Table 1: PCG for two-dimensional diffusion

The truncation precision $\varepsilon_{\mathcal{H}}$, see the beginning of Section 4, has to be decreased if n increases. This is mainly due to the condition number which grows with n, see (19). Apparently, the presented preconditioner is able to adapt itself to the varying coefficients. The dependence of the computational effort on a is surprisingly weak.

5.2 Convection-diffusion problems

In the next test operators of type

$$L = -\Delta + c \cdot \nabla$$

will be considered. The convection coefficient c is randomly chosen, i.e., $c(x) \in [-a, a]^2$ for $x \in \Omega := (0,1)^2$. Table 2 shows the number of iterations for different parameters a. Note that in this case a symmetry of the stiffness matrix cannot be exploited. Therefore, BiCGstab was used as a solver. For the truncation accuracy $\varepsilon_{\mathcal{H}} = 0.2$ was chosen independently of n

	$a = 10$				$a = 100$			
n	T_{LU}	MB	Its	T_{It}	T_{LU}	MB	Its	T_{It}
39 061	3.4s	55.5	10	0.8s	3.5s	55.7	8	0.6s
78 961	7.1s	112.6	16	2.8s	7.0s	112.4	15	2.6s
159 201	19.4s	242.0	19	8.9s	19.5s	241.8	20	8.6s
318 096	40.2s	489.8	20	21.9s	39.6s	489.7	25	21.8s
638 401	106.2s	1045.3	32	64.5s	105.6s	1045.3	34	81.2s

Table 2: preconditioned BiCGstab for diffusion-convection problems

Although the computational effort has increased compared with the diffusion problem, it still scales almost linearly. A dependence on the coefficient c can hardly be observed. We could only test relatively small amplitudes a since the finite element discretization becomes unstable in the convection-dominated case.

5.3 Three-dimensional diffusion

As we have seen in Section 2.1, the structure of $\mathcal{H}$-matrices can equally be applied to any quasi-uniform finite element discretization of Ω given just by the grid information. In order to demonstrate that the $\mathcal{H}$-*LU* decomposition is also efficient for three-dimensional problems, we test the proposed preconditioner on two tetrahedral discretizations ($n = 34403$ and $n = 298727$ of unknowns) of the volume shown in Figure 4. The meshes were generated using NETGEN [21].

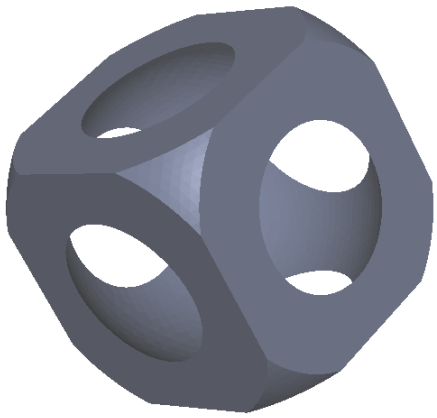


Figure 4: The computational domain

We consider the Dirichlet boundary value problem

$$\begin{aligned} -\operatorname{div} A(x)\nabla u &= 0 \quad \text{in } \Omega, \\ u &= f \quad \text{on } \partial\Omega, \end{aligned}$$

where $A(x) \in \mathbb{R}^{3\times 3}$ is a symmetric positive definite matrix for all $x \in \Omega$. The coefficients a_{ij}, $i, j = 1, 2, 3$, are set to one in the left half space and to a random number from the interval $[0, a]$ in the right half space.

In Table 3 we compare the numerical effort to generate a preconditioner based on the hierarchical Cholesky decomposition. The minimal cluster size was chosen to be $n_{\min} = 50$. For increasing amplitudes a in the second column the time that was needed to compute the approximate Cholesky decomposition is shown. The memory consumption can be found in the third column. Column four and five contain the number of iterations and the CPU time required for PCG to reach a relative accuracy of the residual below 10^{-4}. For these test $\varepsilon_{\mathcal{H}} = 0.8$ was chosen. Compared with the two-dimensional problems, the CPU time for generating the preconditioner

	$n = 34403$				$n = 298727$			
	T_{LL^T}	MB	Its	T_{It}	T_{LL^T}	MB	Its	T_{It}
$a = 10^3$	0.6s	15.2	19	0.5s	17.3s	190.6	34	11.1s
$a = 10^6$	0.6s	15.2	20	0.5s	17.3s	190.0	34	11.0s
$a = 10^9$	0.6s	15.2	20	0.5s	17.3s	190.1	34	11.1s

Table 3: PCG for three-dimensional diffusion

has increased. However, a similar behavior as for the two-dimensional tests can be observed: The

number of iterations of PCG is almost constant. The proposed preconditioner is able to adapt itself to the varying coefficients. The numerical effort scales almost linearly and a dependence on the coefficients can hardly be observed.

6 Conclusion

The convergence rate of iterative solvers for large sparse linear systems stemming from the discretization of elliptic differential boundary value problems suffers from non-smooth coefficients in the operator. Direct methods are robust but due to fill-in lead to non-competitive complexity orders. In this article we have presented an existence result for the $\mathcal{H}$-matrix approximation of the LU decomposition of finite element Galerkin matrices. This approximate LU decomposition can be computed with almost linear complexity while keeping the robustness of the pointwise LU decomposition. Low-precision approximants can be used for preconditioning iterative solvers. The proposed preconditioner is able to achieve problem-independent convergence rates.

References

[1] P. Amestoy, I. S. Duff, J.-Y. L'Excellent, and J. Koster: *A fully asynchronous multifrontal solver using distributed dynamic scheduling.* SIAM J. Matrix Anal. Appl. 23, 15, 2001.

[2] M. Bebendorf: *Efficient inversion of Galerkin matrices of general second order elliptic differential operators.* Preprint 6/2004, Max-Planck-Institut MIS, Leipzig; to appear in Math. Comp.

[3] M. Bebendorf: *Effiziente numerische Lösung von Randintegralgleichungen unter Verwendung von Niedrigrang-Matrizen.* dissertation.de, Verlag im Internet, 2001. ISBN 3-89825-183-7.

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

[5] M. Bebendorf and S. Rjasanow: *Adaptive Low-Rank Approximation of Collocation Matrices.* Computing 70, 1–24, 2003.

[6] M. Bebendorf and W. Hackbusch: *Existence of $\mathcal{H}$-Matrix Approximants to the Inverse FE-Matrix of Elliptic Operators with L^∞-Coefficients.* Numer. Math. 95, 1–28, 2003.

[7] M. Bebendorf: *Approximate Inverse Preconditioning of FE Systems for Elliptic Operators with non-smooth Coefficients.* Preprint 7/2004, Max-Planck-Institute for Mathematics in the Sciences, Leipzig.

[8] J. H. Bramble, J. E. Pasciak, and J. Xu: *Parallel multilevel preconditioners.* Math. Comp. 55, 1–22, 1990.

[9] J. W. Demmel and N. J. Higham and R. Schreiber: *Stability of block LU factorization.* Numer. Linear Algebra Appl. 2, 173–190, 1995.

[10] L. Grasedyck: *Theorie und Anwendungen Hierarchischer Matrizen.* Dissertation, Universität Kiel, 2001.

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

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

[13] W. Hackbusch: *Multi-Grid Methods and Applications.* Springer, 1985.

[14] W. Hackbusch: *Theorie und Numerik elliptischer Differentialgleichungen.* B. G. Teubner, Stuttgart, 1996 – English translation: *Elliptic differential equations. Theory and numerical treatment.* Springer-Verlag, Berlin, 1992.

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

[16] N. J. Higham: *Accuracy and stability of numerical algorithms.*, Second Edition, SIAM, Philadelphia, PA, 2002.

[17] M. Lintner: *Lösung der 2D Wellengleichung mittels hierarchischer Matrizen.* Technische Universität München, Germany, 2002.

[18] V. Rokhlin: *Rapid solution of integral equations of classical potential theory.* J. Comput. Phys. 60:187–207, 1985.

[19] J. W. Ruge and K. Stüben: *Algebraic multigrid.* in Multigrid Methods, edited by S. F. McCormick, p. 73, SIAM, 1987.

[20] Y. Saad: *Iterative Methods for Sparse Linear Systems.* PWS Publishing, Boston, 1996.

[21] J. Schöberl: *NETGEN – An advancing front 2D/3D-mesh generator based on abstract rules.* Comput. Visual. Sci., 1:41–52, 1997.

[22] B. F. Smith, P. E. Bjørstad, and W. D. Gropp: *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations.* Cambridge Univ. Press, 1996.

[23] E. Tyrtyshnikov: *Mosaic-skeleton approximations.* Calcolo 33, 47–57 (1998), 1996.