A preconditioner based on low-rank approximation of Schur complements

P. Gatto[a,*], J.S. Hesthaven[a]

[a]*MATHICSE, Ecole Polytechnique Fédérale de Lausanne (EPFL), MA C2 652 (Batiment MA), Station 8, CH-1015 Lausanne, Switzerland*

Abstract

We introduce a preconditioner based on low-rank compression of Schur complements. The construction is inspired by standard nested-dissection and relies on the assumption that the Schur complements can be approximated to high precision by Hierarchical-Block-Separable matrices. We build the preconditioner as an approximate LDM^t factorization of a given matrix A, and no knowledge of A in assembled form is required by the construction. The LDM^t is amenable to fast inversion and the inverse can be applied fast as well. We investigate the behavior of the preconditioner in the context of DG finite element approximations of elliptic and hyperbolic problems.

Keywords: Preconditioned GMRES, Interpolative Decomposition

1. Introduction

This work rests on the observation that, for a large class of problems, the dense Schur complement matrices that arise in the nested dissection method are rank-structured, thus allowing for the use of accelerated matrix algebra, see, e.g., [4, 13]. In the case of well-behaved elliptic problems, this property is a consequence of the rapid decay of the underlying Green function. To the contrary, in the case of wave-propagation problems, the same argument does not apply, and the reasons behind the

*Corresponding author

Email addresses: paolo.gatto@epfl.ch (P. Gatto), jan.hesthaven@epfl.ch (J.S. Hesthaven)

rank-structure of the Schur complements remain poorly understood. Nevertheless, by exploiting this property, approximate matrix decompositions can be constructed cheaply, and turn out to be excellent preconditioners.

Linear systems that arise from finite elements discretizations of wave propagation phenomena are typically poorly conditioned and highly indefinite. Consequently, it is both vital and challenging to construct an effective preconditioner. Standard multigrid methods generally fail to carry the oscillations at the wavelength scale onto the coarse grids. Incomplete LU decompositions require to assemble the global matrix, are fairly expensive to compute, and still lead to a number of iterations that is frequency-dependent. The lack of effective preconditioning techniques has lead to the use of (sparse) direct solvers. Fast methods, such as the fast-multipole-method, are confined to problems for which the governing PDE's can be reformulated as boundary integral equations (BIE's). The linear systems resulting from the discretization of the BIE's are dense, as opposed to sparse, and often well-conditioned. Over the last twenty years, a number of methods has been developed for their efficient solution. They are based on a rigorous understanding of the physics of the problem, along with sophisticated analytical arguments that rely on asymptotic expansions of special functions. As a result, they have limited applicability, e.g., variable coefficients problems are out of reach, and their efficient implementation remains quite challenging.

As of today, because of the lack of robust preconditioners, discretizations of wave propagation problems have eluded the reach of fast iterative solvers. In this work, we introduce a preconditioner whose construction is completely general, is parallel in nature, and fits modern computational architectures. Results obtained in the context of Discontinuous Garlerkin (DG) finite elements approximations show that the behavior of the preconditioner, measured as the number of GMRES iterations, is independent of both the mesh size and the order of approximation.

The paper is organized as follows. In Section 2 we review the concept of Hierarchically-Block-Separable matrices, and describe how a matrix can be reduced to such form within linear complexity. In Section 3 we develop analytical arguments to justify the low-rank nature of the Schur complements. The construction of the preconditioner is described in Section 4, and numerical examples are reported in Section 5. Finally, in Section 6, we draw conclusions from this work, and point towards future directions of research.

2. Hierarchical-Block-Separable Matrices

In vague terms, a matrix is Hierarchical-Block-Separable (HBS) if its off-diagonal blocks admit a low-rank factorization, and such factors satisfy certain recursion relations that make the matrix inexpensive to store and manipulate. More precisely, if k is the off-diagonal rank of a square HBS matrix of size N, then such matrix can be applied to a vector and inverted in $\mathrm{O}(Nk)$ and $\mathrm{O}(Nk^2)$ operations. respectively. The following presentation closely follows the one in [7].

Let A be a $2^L m \times 2^L m$ matrix and partition its index vector $I = (1, \ldots, 2^L m)$ recursively through a binary tree. For each tree level $\ell \in \{1, \ldots, L\}$, we obtain a tessellation of A into $2^\ell \times 2^\ell$ square blocks of size $2^{L-\ell} m$, see Figure 1. Tree nodes that belong to the finest level, namely $\ell = L$, will occasionally be called leaf nodes. Finally, for every node σ on level ℓ, there exists a unique node τ on the same level such that σ and τ have a common ancestor on level $\ell - 1$. We call $\{\sigma, \tau\}$ a sibling pair.

Matrix A is an *$\mathcal{S}$-matrix* or a *semi-separable* matrix if there exists an integer k such that, for every sibling pair $\{\sigma, \tau\}$ of the tree, the off-diagonal blocks $A(\sigma, \tau)$[1] and $A(\tau, \sigma)$ have (numerical) ranks equal to k. Consequently, we can factor the off-diagonal blocks as:

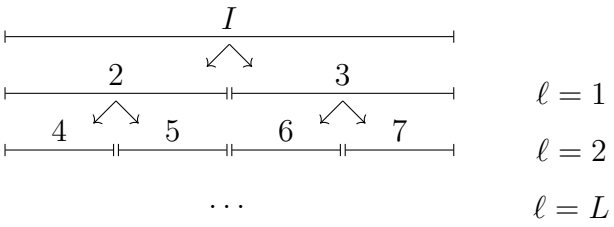
$$A(\sigma, \tau) = U_\sigma^{\text{tall}} \, \tilde{A}_{\sigma,\tau} \left(V_\tau^{\text{tall}}\right)' \qquad ; \qquad A(\tau, \sigma) = U_\tau^{\text{tall}} \, \tilde{A}_{\tau,\sigma} \left(V_\sigma^{\text{tall}}\right)'$$

where matrices $\tilde{A}_{\sigma,\tau}$ and $\tilde{A}_{\tau,\sigma}$ have size $k \times k$, which is independent of ℓ. When $\{\sigma, \tau\}$ belongs to level ℓ, the tall (and hopefully skinny!) matrices U_σ^{tall}, U_τ^{tall}, V_σ^{tall}, V_τ^{tall} have size $2^{L-\ell} m \times k$. The assumption that the (numerical) rank of sibling interactions is constant across the entire tree is, in practice, replaced by an assumption of boundedness see Section **??** and Section 5 for further discussion.

Let us define the block-diagonal matrices:

$$\begin{aligned}
D^{(L)} &= \text{diag}\{D_\sigma = A(\sigma, \sigma) \;\; : \;\; \tau \text{ belongs to level } L\} \\
U_{\text{tall}}^{(\ell)} &= \text{diag}\{U_\sigma^{\text{tall}} \;\; : \;\; \sigma \text{ belongs to level } \ell\} && \text{for } \ell = 1, \ldots, L \\
V_{\text{tall}}^{(\ell)} &= \text{diag}\{V_\sigma^{\text{tall}} \;\; : \;\; \sigma \text{ belongs to level } \ell\} && \text{for } \ell = 1, \ldots, L
\end{aligned}$$

[1] Throughout the paper, the MATLAB®-like notation $A(\sigma, \tau)$ indicates the restriction of A to row-index vector σ and column-index vector τ.



(a) Partition of index vector I into a binary tree

$A =$

(4,4)	(4,5)	(2,3)	
(5,4)	(5,5)		
(3,2)		(6,6)	(6,7)
		(7,6)	(7,7)

(b) Blocks of matrix A corresponding to binary tree partition of index vector I.

Figure 1:

and let $\tilde{A}^{(\ell)}$ be the size $2^\ell k \times 2^\ell k$ block-matrix comprised of all blocks $\tilde{A}_{\sigma,\tau}$ such that $\{\sigma, \tau\}$ are a sibling pair on level ℓ. The $\mathcal{S}$-matrix A can be factorized as follows:

$$A = U^{(1)}_{\text{tall}} \tilde{A}^{(1)} \left(V^{(1)}_{\text{tall}}\right)' + \cdots + U^{(L)}_{\text{tall}} \tilde{A}^{(L)} \left(V^{(L)}_{\text{tall}}\right)' + D^{(L)} \tag{1}$$

A pictorial description of the factorization is shown in Figure 2. Informally speaking, the first term of the sum is the level-1 off-diagonal part of A, the second term is the level-2 (remaining) off-diagonal part, and so on up to the finest level L. Matrix $\tilde{A}^{(\ell)}$ is a $2^\ell k \times 2^\ell k$ block-tridiagonal matrix, with null blocks on the diagonal.

$\mathcal{S}$-matrices require to store and manipulate "tall" matrices at all tree-levels. This undesirable property can be circumvented by imposing one additional condition on the U- and V-factors. An $\mathcal{S}$-matrix A is said to be *Hierarchically-Block-Separable* (HBS) if it satisfies:

$$U^{\text{tall}}_\sigma = \begin{pmatrix} U^{\text{tall}}_\mu & \\ & U^{\text{tall}}_\nu \end{pmatrix} U_\sigma \qquad ; \qquad V^{\text{tall}}_\sigma = \begin{pmatrix} V^{\text{tall}}_\mu & \\ & V^{\text{tall}}_\nu \end{pmatrix} V_\sigma$$

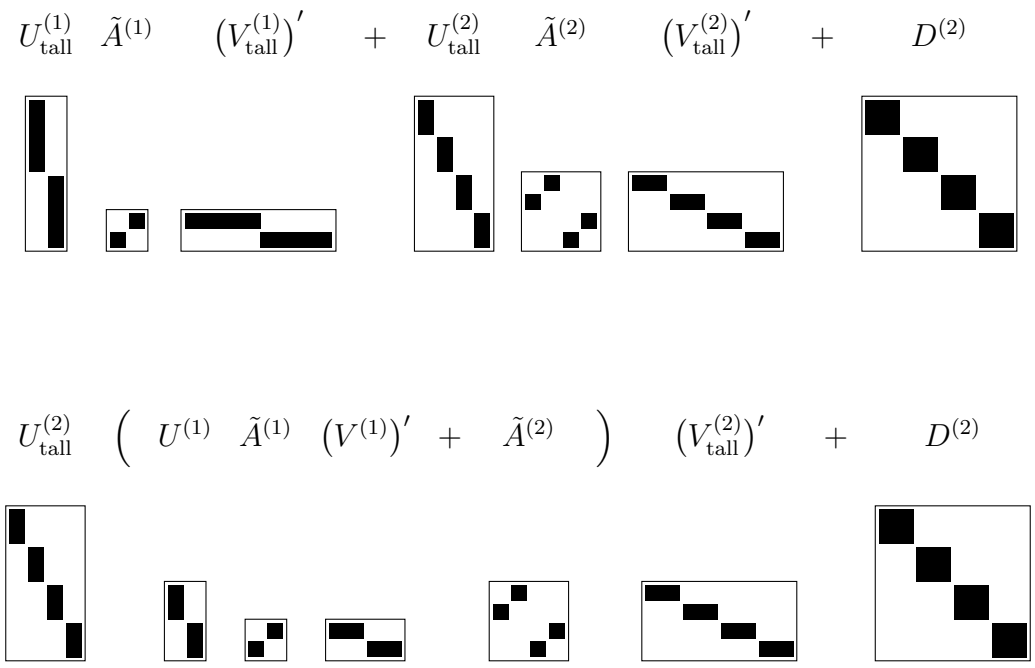


Figure 2: Factorization of an $\mathcal{S}$-matrix (above), as opposed to the factorization of an HBS matrix (below), for $L = 2$.

for every σ belonging to tree-level $\ell = 1, \ldots, L-1$, with descending sibling pair $\{\mu, \nu\}$. This is essentially a condition about nesting of column and row spaces.

All matrices U_σ and V_σ have size $2k \times k$. When they are collected into block-diagonal matrices

$$\begin{aligned} U^{(\ell)} &= \text{diag}\{U_\sigma \ : \ \sigma \text{ belongs to level } \ell\} \qquad \text{for } \ell = 1, \ldots, L-1 \\ V^{(\ell)} &= \text{diag}\{V_\sigma \ : \ \sigma \text{ belongs to level } \ell\} \qquad \text{for } \ell = 1, \ldots, L-1 \end{aligned}$$

the following recursions are obtained:

$$U^{(\ell)}_{\text{tall}} = U^{(L)}_{\text{tall}}\, U^{(L-1)} \cdots U^{(\ell)} \quad , \quad V^{(\ell)}_{\text{tall}} = V^{(L)}_{\text{tall}}\, V^{(L-1)} \cdots V^{(\ell)} \qquad \text{for } \ell = 1, \ldots, L-1$$

Thus, in the case of an HBS matrix, factorization (1) simplifies to:

$$\begin{aligned}
A &= U_{\text{tall}}^{(L)} \cdots U^{(1)} \, \tilde{A}^{(1)} \, V^{(1)\prime} \cdots V_{\text{tall}}^{(L)\prime} + \cdots + U_{\text{tall}}^{(L)} \, \tilde{A}^{(L)} \, V_{\text{tall}}^{(L)\prime} + D^{(L)} \\
&= U_{\text{tall}}^{(L)} \Big(U^{(L-1)} \cdots U^{(1)} \, \tilde{A}^{(1)} \, V^{(1)\prime} \cdots V^{(L-1)\prime} + \cdots + \tilde{A}^{(L)} \Big) V_{\text{tall}}^{(L)\prime} + D^{(L)} \\
&= U_{\text{tall}}^{(L)} \Big(U^{(L-1)} \Big(U^{(L-2)} \cdots U^{(1)} \, \tilde{A}^{(1)} \, V^{(1)\prime} \cdots V^{(L-2)\prime} + \cdots + \tilde{A}^{(L-1)} \Big) V^{(L-1)} \\
&\qquad\qquad + \tilde{A}^{(L)} \Big) V_{\text{tall}}^{(L)\prime} + D^{(L)} \\
&= \cdots
\end{aligned}$$

where we have exposed the recursive nature of the factorization.

Rather than HBS matrices *per se*, the construction of HBS approximants is of practical interest. More precisely, given a matrix A and a tolerance ε, we seek an HBS-matrix $A^{(\text{HBS})}$ such that $\|A - A^{(\text{HBS})}\| \leq \varepsilon$, for some suitable norm $\|\cdot\|$. If A is an arbitrary square matrix of size N, a straightforward approach for computing $A^{(\text{HBS})}$ yields a $\mathrm{O}(kN^2)$ cost, where k is the HBS-rank. In practice, this is prohibitively expensive. Nevertheless, under moderate assumptions on A, it is possible to construct $A^{(\text{HBS})}$ at the acceptable cost of $\mathrm{O}(Nk^2)$, see [10]. Thus, provided that k is independent of N^2, A can be reduced to HBS-form within linear complexity. Let us recall the exact result.

Theorem 2.1. Let A be an $N \times N$ hierarchical-block-separable matrix that has HBS-rank k. Suppose that:

1. matrix-vector products $x \mapsto Ax$ and $x \mapsto A^t x$ can be evaluated at a cost T_{mult};
2. individual entries of A can be evaluated at a cost T_{entry}.

An HBS factorization of A can be computed in a time proportional to

$$T_{\text{mult}} \times 2(k+p) + T_{\text{rand}} \times N(k+p) + T_{\text{entry}} \times 2Nk + T_{\text{flop}} \times cNk^2$$

where T_{rand} is the time required to generate a random number, T_{flop} is the time required to perform a floating point operation, p is a small oversampling parameter (typically $p = 10$), and c is a small constant independent of N or k. ∎

[2] As we shall see in Section 5, this is never the case in practice, and some dependency of k upon N is to be expected.

The construction described in [10] relies heavily on the nesting of the basis of the off-diagonal blocks, and on the compression of such blocks through Interpolative Decompositions (ID), see, e.g., Section 4 and 5 of [5] for a detailed discussion on interpolative decompositions. In general terms, if A is an $m \times n$ matrix, an ID is a factorization of the form:

$$A = A^{(\text{skel})} P$$

where $A^{(\text{skel})}$ is a "skeletonization" of A, namely it is an $m \times k$ matrix constructed by selecting k columns of A, and P is a well-behaved[3] $k \times n$ matrix that, obviously, contains an identity of size k. The cost of computing a rank-k ID of A is $\mathrm{O}\big(mkn \log(n)\big)$. As shown in [12], this cost can be lowered to $\mathrm{O}\big(mn \log(l) + lkn \log(n)\big)$, where l is an integer greater than but close to k (in applications, $l = k + 10$ is typical.) The first term is the cost of obtaining l samples of the range of A *via* an accelerated fast Fourier transform. Consequently, if we additionally assume that A can be applied to a vector within linear complexity, we can drop the first term, and the cost of computing the ID reduces to

$$\mathrm{O}\big(lkn \log(n)\big) \tag{3}$$

This is the cornerstone to the establish the complexity of the algorithm on which rests the proof of Theorem 2.1.

3. Analytical Apparatus

4. Preconditioner Construction

In this section we describe a variant of the well-known nested dissection algorithm introduced by George in [6]. Our construction extends the work of Gilmann and Martisson, see [8], developed in the context of finite difference approximations of elliptic PDE's. Instead of geometrical considerations similar to domain decomposition techniques, we rely on a purely algebraic, black-box approach that allows us to handle a much more general framework. In fact, we shall only require that A is a sparse matrix arising from a discretization of a differential operator. Although, in practice, all other known properties of A could—and should!—be exploited, such properties should, in principle, affect the performance of the preconditioner, not its construction.

[3] In the present context, by "well-behaved", we refer to the fact that no entry of P has absolute value greater than 2. Let us recall that in general it is possible to obtain an ID where the entries of P are bounded in absolute valued by 1, although this is an NP-hard problem.

Our approach rests upon a recursive reordering of the degrees of freedom (dof's) of A, which generates a binary tree. Such reordering dictates a hierarchy of Schur complements, in the sense that the complement associated to a node is constructed by "merging" those associated to its descendants on the lower level. The novelty is that, provided that A is sparse and that all Schur complements are to high precision Hierarchically-Block-Separable, the following facts hold true:

1. an LDM^t factorization of A can be realized within linear complexity;
2. the factorization can be inverted within linear complexity;
3. the inverse can be applied within linear complexity.

The factorization can be realized with a trivially parallel process, whose cost is dominated by the cost of processing the Schur complements on the top tree level. The fact that the inverse factorization can be applied fast makes it perfectly suitable to be employed as a preconditioner.

4.1. Matrix Reordering

Let us partition the dof's into two boxes, $\mathrm{Box}_1, \mathrm{Box}_2$, and, for each box Box_i, identify interior dof's I^i and boundary dof's B^i in the sense of the following connectivity graph:

$$\begin{array}{ccc} B^1 & \longleftrightarrow & B^2 \\ \updownarrow & & \updownarrow \\ I^1 & & I^2 \end{array} \tag{4}$$

The interpretation is straightforward: distinct boxes are connected to each other, in the sense of an algebraic graph, through their boundaries dof's only. At this level of exposition, the most attractive partition is the one that maximizes the number of interior dof's while minimizing the number of boundary dof's. The construction proceeds by repartitioning each box into a pair of sibling boxes, in fact creating a binary tree of boxes[4], see Figure 3, and by identifying boundary and interior dof's, in the sense of graph (4), for each new pair of sibling boxes. For illustration, the

[4]Strictly speaking, this is a tree with missing route, i.e., a forest. In fact, if we were to introduce a single top-level box holding the entirety of the dof's, under a purely algebraic approach, no boundary dof's could be identified.

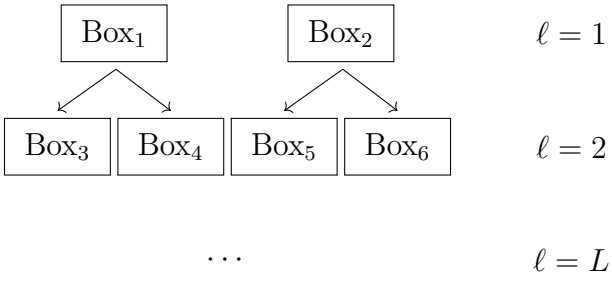
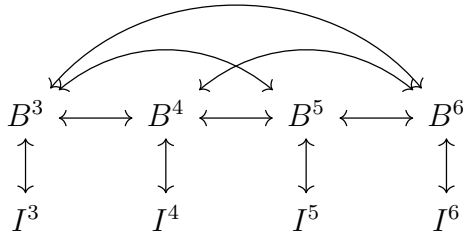


Figure 3: Binary tree of boxes for dof's partition

connectivity graph of boxes on the second tree level, i.e., $\ell = 2$, is:



The construction terminates at tree level $\ell = L$, when the newly created boxes contain a number of dof's sufficiently small to allow for dense linear algebra operations at a negligible cost. For consistency with the notation introduced in Section 2, we switch to Greek letters and identify a box with its index, i.e., $\sigma = \text{Box}_\sigma$.

Let us establish an ordering for the dof's in I^σ and B^σ, and define the following sub-matrices of A:

$A^{(\sigma)}{}_{ii} = A(I^\sigma, I^\sigma)$	σ-box interior-to-interior
$A^{(\sigma)}{}_{bb} = A(B^\sigma, B^\sigma)$	σ-box boundary-to-boundary
$A^{(\sigma)}{}_{bi} = A(B^\sigma, I^\sigma)$	σ-box boundary-to-interior
$A^{(\sigma)}{}_{ib} = A(I^\sigma, B^\sigma)$	σ-box interior-to-boundary
$A^{(\sigma,\tau)} = A(B^\sigma, B^\tau)$	σ-box to τ-box (boundary interaction only)

Let $\sigma_1, \ldots, \sigma_n$ be the leaf boxes and order the dof's of A by grouping together the interior dof's $I^{\sigma_1}, \ldots, I^{\sigma_n}$, and the boundary dof's $B^{\sigma_1}, \ldots, B^{\sigma_n}$ on tree level $\ell = L$.

Then A has the following block-structure:

$$A = \left(\begin{array}{ccc|ccc} {A^{(\sigma_1)}}_{ii} & & & {A^{(\sigma_1)}}_{ib} & & \\ & \ddots & & & \ddots & \\ & & {A^{(\sigma_n)}}_{ii} & & & {A^{(\sigma_n)}}_{ib} \\ \hline {A^{(\sigma_1)}}_{bi} & & & {A^{(\sigma_1)}}_{bb} & \cdots & {A^{(\sigma_1,\sigma_n)}} \\ & \ddots & & \vdots & \ddots & \vdots \\ & & {A^{(\sigma_n)}}_{bi} & {A^{(\sigma_n,\sigma_1)}} & \cdots & {A^{(\sigma_n)}}_{bb} \end{array}\right) \tag{5}$$

As common practice, we omit null blocks and use $\star$'s to indicate non-zero blocks. The matrix partitioning indicates the so-called super-blocks, i.e., blocks that emerge from a super-partition of the dof's into all interior and boundary dof's for a particular tree level.

4.2. Hierarchical Matrix Factorization

The factorization strategy recursively decouples interior dof's from boundary dof's through Gauss transforms, starting from the leaf-boxes, all the way up to the top tree-level. The Schur complements that arise in the process are treated through accelerated linear algebra techniques. More specifically, the complement associated to a box is obtained by a fast merge of the complements of its child-boxes.

For each box σ, we define Guass transforms $L^{(\sigma)}$ and $M^{(\sigma)}$ as unit lower triangular matrices such that:

$$L^{(\sigma)}(B^\sigma, I^\sigma) = {A^{(\sigma)}}_{bi}\,{{A^{(\sigma)}}_{ii}}^{-1} \qquad , \qquad M^{(\sigma)}(B^\sigma, I^\sigma) = \left({{A^{(\sigma)}}_{ii}}^{-1}\,{A^{(\sigma)}}_{ib}\right)^t \tag{6}$$

Since it is well-understood how Guass transforms accumulate and commute with permutations, we refer to them generically as L and M. The meaning of each instance can be inferred by the context. Through Gauss transforms, we decouple the top-left

super-block of A:

$$L^{-1}AM^{-t} = \left(\begin{array}{ccc|cccc} {A^{(\sigma_1)}}_{ii} & & & & & & \\ & \ddots & & & & & \\ & & {A^{(\sigma_n)}}_{ii} & & & & \\ \hline & & & S^{(\sigma_1)} & A^{(\sigma_1,\sigma_2)} & \cdots & A^{(\sigma_1,\sigma_n)} \\ & & & A^{(\sigma_2,\sigma_1)} & \ddots & \ddots & \vdots \\ & & & \vdots & \ddots & \ddots & A^{(\sigma_{n-1},\sigma_n)} \\ & & & A^{(\sigma_n,\sigma_1)} & \cdots & A^{(\sigma_n,\sigma_{n-1})} & S^{(\sigma_n)} \end{array}\right) \quad (7)$$

The top-right and bottom-left super-blocks vanish, while Schur complements $S^{(\sigma)} = {A^{(\sigma)}}_{bb} - {A^{(\sigma)}}_{bi}\,{A^{(\sigma)}}_{ii}^{-1}\,{A^{(\sigma)}}_{ib}$ appear on the diagonal of the bottom-right super-block.

In order to recursively proceed in the factorization, for each pair of sibling boxes $\{\mu,\nu\}$ with common ancestor σ, we define the remaining interior dof's $\hat{I}^\sigma = (B^\mu \cup B^\nu) \cap I^\sigma$ and the remaining boundary dof's $\hat{B}^\sigma = (B^\mu \cup B^\nu) \cap B^\sigma$. Consequently, $\{\hat{I}^\sigma, \hat{B}^\sigma\}$ is a partition of the aggregated boundary $B^\mu \cup B^\nu$, while $\{B^\mu \cap \hat{I}^\sigma, B^\mu \cap \hat{B}^\sigma\}$ is a partition[5] of B^μ. Up to a permutation we shall omit, we partition the Schur complement $S^{(\mu)}$ as:

$$S^{(\mu)} = \begin{pmatrix} {S^{(\mu)}}_{ii} & {S^{(\mu)}}_{ib} \\ {S^{(\mu)}}_{bi} & {S^{(\mu)}}_{bb} \end{pmatrix}$$

where the blocks are defined as:

$$\begin{aligned} {S^{(\mu)}}_{ii} &= S^{(\mu)}(B^\mu \cap \hat{I}^\sigma, B^\mu \cap \hat{I}^\sigma) \\ {S^{(\mu)}}_{bb} &= S^{(\mu)}(B^\mu \cap \hat{B}^\sigma, B^\mu \cap \hat{B}^\sigma) \\ {S^{(\mu)}}_{bi} &= S^{(\mu)}(B^\mu \cap \hat{B}^\sigma, B^\mu \cap \hat{I}^\sigma) \\ {S^{(\mu)}}_{ib} &= S^{(\mu)}(B^\mu \cap \hat{I}^\sigma, B^\mu \cap \hat{B}^\sigma) \end{aligned}$$

Let us define matrices:

$${\hat{A}^{(\sigma)}}_{ii} = \begin{pmatrix} {S^{(\mu)}}_{ii} & \hat{A}^{(\mu,\nu)} \\ \hat{A}^{(\nu,\mu)} & {S^{(\nu)}}_{ii} \end{pmatrix} ; {\hat{A}^{(\sigma)}}_{ib} = \begin{pmatrix} {S^{(\mu)}}_{ib} & \hat{A}^{(\mu,\nu)} \\ \hat{A}^{(\nu,\mu)} & {S^{(\nu)}}_{ib} \end{pmatrix} ; {\hat{A}^{(\sigma)}}_{bb} = \begin{pmatrix} {S^{(\mu)}}_{bb} & \hat{A}^{(\mu,\nu)} \\ \hat{A}^{(\nu,\mu)} & {S^{(\nu)}}_{bb} \end{pmatrix} \quad (8)$$

[5]The two sets are evidently disjoint. Furthermore: $(B^\mu \cap \hat{I}^\sigma) \cup (B^\mu \cap \hat{B}^\sigma) = B^\mu \cap (\hat{I}^\sigma \cup \hat{B}^\sigma) = B^\mu \cap (B^\mu \cup B^\nu) = B^\mu$.

and $\hat{A}^{(\sigma)}{}_{bi}$ analogously to $\hat{A}^{(\sigma)}{}_{ib}$. In order to simplify the notation, $\hat{A}^{(\mu,\nu)}$ indicates a sub-matrix of $A^{(\mu,\nu)}$ that can be inferred from the context[6]. By reordering the boundary dof's $B^{\sigma_1}, \ldots, B^{\sigma_n}$ as $\hat{I}^{(\cdot)}, \ldots, \hat{I}^{(\cdot)}, \hat{B}^{(\cdot)}, \ldots, \hat{B}^{(\cdot)}$, where the omitted superscripts are the parent boxes of the leaves $\sigma_1, \ldots, \sigma_n$, equation (7) becomes:

$$L^{-1}AM^{-t} = \left(\begin{array}{c|ccccc} \star & & & & & \\ \hline & \hat{A}^{(\cdot)}{}_{ii} & & & \hat{A}^{(\cdot)}{}_{ib} & & \\ & & \ddots & & & \ddots & \\ & & & \hat{A}^{(\cdot)}{}_{ii} & & & \hat{A}^{(\cdot)}{}_{ib} \\ & \hat{A}^{(\cdot)}{}_{bi} & & & \hat{A}^{(\cdot)}{}_{bb} & \cdots & \hat{A}^{(\cdot,\cdot)} \\ & & \ddots & & \vdots & \ddots & \vdots \\ & & & \hat{A}^{(\cdot)}{}_{bi} & \hat{A}^{(\cdot,\cdot)} & \cdots & \hat{A}^{(\cdot)}{}_{bb} \end{array}\right)$$

The key observation is that the block-structure of the bottom-right super-block is identical to that of the original matrix A, as shown in (5). The interior dof's $\hat{I}^{(\cdot)}, \ldots, \hat{I}^{(\cdot)}$ are recursively decoupled through Gauss transforms until the top tree-level is reached. The Schur complement that originates for the elimination of the interior dof's $\hat{I}^{\sigma}$ has the following structure:

$$S^{(\sigma)} = \underbrace{\begin{pmatrix} S^{(\mu)}{}_{bb} & \hat{A}^{(\mu,\nu)} \\ \hat{A}^{(\nu,\mu)} & S^{(\nu)}{}_{bb} \end{pmatrix}}_{\hat{A}^{(\sigma)}{}_{bb}} - \underbrace{\begin{pmatrix} S^{(\mu)}{}_{bi} & \hat{A}^{(\mu,\nu)} \\ \hat{A}^{(\nu,\mu)} & S^{(\nu)}{}_{bi} \end{pmatrix}}_{\hat{A}^{(\sigma)}{}_{bi}} \underbrace{\begin{pmatrix} S^{(\mu)}{}_{ii} & \hat{A}^{(\mu,\nu)} \\ \hat{A}^{(\nu,\mu)} & S^{(\nu)}{}_{ii} \end{pmatrix}^{-1}}_{\hat{A}^{(\sigma)}{}_{ii}} \underbrace{\begin{pmatrix} S^{(\mu)}{}_{ib} & \hat{A}^{(\mu,\nu)} \\ \hat{A}^{(\nu,\mu)} & S^{(\nu)}{}_{ib} \end{pmatrix}}_{\hat{A}^{(\sigma)}{}_{ib}} \tag{9}$$

At the end of the decoupling process, we obtain the following factorization:

$$L^{-1}AM^{-t} = \begin{pmatrix} \hat{A}^{(\sigma_1)}{}_{ii} & & & & \\ & \ddots & & & \\ & & \hat{A}^{(\sigma_n)}{}_{ii} & & \\ & & & S^{(1)} & A^{(1,2)} \\ & & & A^{(2,1)} & S^{(2)} \end{pmatrix} \tag{10}$$

where the boxes $\sigma_1, \ldots, \sigma_n$ are ordered starting from the bottom tree-level and, for consistency of notation, we have set $\hat{A}^{(\sigma)}{}_{ii} = A^{(\sigma)}{}_{ii}$ for the leaf-boxes as well. The matrices L and M are the accumulated Gauss transforms[7], relative to all boxes,

[6] Distinct instances of the symbol should be regarded as different matrices.

[7] As before, permutations have been omitted in the definition of the Gauss transforms.

excluding those on the top tree-level:

$$L = L^{(\sigma_1)} \cdots L^{(\sigma_n)} \qquad ; \qquad M = M^{(\sigma_1)} \cdots M^{(\sigma_n)}$$

Finally, if we define:

$$\hat{A}^{(0)} = \begin{pmatrix} S^{(1)} & A^{(1,2)} \\ A^{(2,1)} & S^{(2)} \end{pmatrix} \qquad ; \qquad D = \mathrm{diag}\{\hat{A}^{(\sigma_1)}{}_{ii}, \ldots, \hat{A}^{(\sigma_n)}{}_{ii}, \hat{A}^{(0)}\} \tag{11}$$

then we obtain the desired LDM^t factorization of A.

4.3. Fast Merging of Schur Complements

The hierarchy of the boxes dictates a hierarchy of the Schur complements, in the sense that $S^{(\sigma)}$ depends only upon the Schur complements of the child-boxes of σ, namely $S^{(\mu)}$ and $S^{(\nu)}$, and some blocks of the original matrix A, see equation (9). Informally, we say that $S^{(\sigma)}$ is obtained by "merging" together $S^{(\mu)}$ and $S^{(\nu)}$. As we are about to show, when $S^{(\mu)}$ and $S^{(\nu)}$ are in HBS-form, the merging procedure can be performed fast, in the sense that an HBS approximation to $S^{(\sigma)}$ can be computed within linear complexity.

The fast merging procedure is based the following assumptions:

1. all sub-matrices $\hat{A}^{(\cdot,\cdot)}$ of A are sparse;
2. all leaf-node Schur complements $S^{(\cdot)}$ and their inverses ${S^{(\cdot)}}^{-1}$ are in HBS-form.

In general, the property of a matrix to be sparse does not carry over to an arbitrarily selected sub-matrix. In the context of high-order finite element approximations, this is well-established. In fact, dense blocks allow advanced finite element solvers to employ high-performance dense linear algebra. We remark that assumption (1) is much milder, since it only requires matrices describing boundary-to-boundary sub-interactions to be sparse.

For ease of exposition, let us recall the definition of $S^{(\sigma)}$ as in (9):

$$S^{(\sigma)} = \underbrace{\begin{pmatrix} S^{(\mu)}{}_{bb} & \hat{A}^{(\mu,\nu)} \\ \hat{A}^{(\nu,\mu)} & S^{(\nu)}{}_{bb} \end{pmatrix}}_{\hat{A}^{(\sigma)}{}_{bb}} - \underbrace{\begin{pmatrix} S^{(\mu)}{}_{bi} & \hat{A}^{(\mu,\nu)} \\ \hat{A}^{(\nu,\mu)} & S^{(\nu)}{}_{bi} \end{pmatrix}}_{\hat{A}^{(\sigma)}{}_{bi}} \underbrace{\begin{pmatrix} S^{(\mu)}{}_{ii} & \hat{A}^{(\mu,\nu)} \\ \hat{A}^{(\nu,\mu)} & S^{(\nu)}{}_{ii} \end{pmatrix}^{-1}}_{\hat{A}^{(\sigma)}{}_{ii}} \underbrace{\begin{pmatrix} S^{(\mu)}{}_{ib} & \hat{A}^{(\mu,\nu)} \\ \hat{A}^{(\nu,\mu)} & S^{(\nu)}{}_{ib} \end{pmatrix}}_{\hat{A}^{(\sigma)}{}_{ib}}$$

Assume that $S^{(\mu)}$ and $S^{(\nu)}$ are in HBS-form. The fast merging procedure is performed through the following steps.

Step 1 Since a sub-matrix of an HBS matrix can be trivially obtained through a tall and skinny permutation matrix P, e.g., ${S^{(\mu)}}_{bb} = P^t\, S^{(\mu)}\, P$, matrices ${\hat{A}^{(\sigma)}}_{bb}$, ${\hat{A}^{(\sigma)}}_{bi}$, and ${\hat{A}^{(\sigma)}}_{ib}$ can be applied to a vector within linear complexity.

Step 2 The action of the inverse of ${\hat{A}^{(\sigma)}}_{ii}$ on a vector z is equivalent to the solution of the linear system:

$$\begin{pmatrix} {S^{(\mu)}}_{ii} & \hat{A}^{(\mu,\nu)} \\ \hat{A}^{(\nu,\mu)} & {S^{(\nu)}}_{ii} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix}$$

where $z = (z_1, z_2)$ and $x = (x_1, x_2)$ have been partitioned according to the blocking of ${\hat{A}^{(\sigma)}}_{ii}$. A standard block-solve yields:

$$x_2 = {\tilde{S}^{(\nu)}}_{ii}^{-1}(z_2 - \hat{A}^{(\nu,\mu)}\, {S^{(\mu)}}_{ii}^{-1}\, z_1) \tag{12a}$$

$$x_1 = {S^{(\mu)}}_{ii}^{-1}\, z_1 - {S^{(\mu)}}_{ii}^{-1}\, \hat{A}^{(\mu,\nu)}\, x_2 \tag{12b}$$

where ${\tilde{S}^{(\nu)}}_{ii} = {S^{(\nu)}}_{ii} - \hat{A}^{(\nu,\mu)}\, {S^{(\mu)}}_{ii}^{-1}\, \hat{A}^{(\mu,\nu)}$. Let us remark that matrices ${S^{(\mu)}}_{ii}^{-1}$ and ${\tilde{S}^{(\nu)}}_{ii}^{-1}$ fully describe the action of ${\hat{A}^{(\sigma)}}_{ii}^{-1}$ through equations (12). Since ${\tilde{S}^{(\nu)}}_{ii}$ can be applied within linear complexity, by virtue of Theorem 2.1, it can be compressed efficiently. The cost of the current step is:

$$\mathrm{O}(\#\hat{I}^\sigma k^2) \quad = \quad \underset{\text{inversion of } {S^{(\mu)}}_{ii}}{\mathrm{O}(\#\hat{I}^\sigma k^2)} \quad + \quad \underset{\text{compression of } {\tilde{S}^{(\nu)}}_{ii}}{\mathrm{O}(\#\hat{I}^\sigma k^2)} \quad + \quad \underset{\text{inversion of } {\tilde{S}^{(\nu)}}_{ii}}{\mathrm{O}(\#\hat{I}^\sigma k^2)}$$

The procedure is detailed in Algorithm (1).

Step 3 Under the assumption that $\#\hat{B}^\sigma = \mathrm{O}(\#\hat{I}^\sigma)$, the previous steps imply that $S^{(\sigma)}$ can be applied to a vector within linear complexity. By Theorem 2.1, it can be reduced to HBS form at the cost $\mathrm{O}(\#\hat{B}^\sigma k^2)$.

Let us remark that, although computing the HBS form of a parent Schur complement is a linear complexity operation, there is not guarantee that it is HBS to high precision. We expect this property to be dictated by the nature of the underlying PDE and the discretization method, and affect the cost/effectiveness ratio of the preconditioner. This is investigated through numerical experiments, that are presented in Section 5.

input : z_1, z_2, $S^{(\mu)}$, $S^{(\nu)}$, $\hat{A}^{(\mu,\nu)}$, $\hat{A}^{(\nu,\mu)}$
output: x_1, x_2, ${S^{(\mu)}}_{ii}^{-1}$, ${\tilde{S}^{(\nu)}}_{ii}^{-1}$

`-- STEP 2.1 --`
determine permutations $P_\mu : B^\mu \to B^\mu \cap \hat{I}^\sigma$, and $P_\nu : B^\nu \to B^\nu \cap \hat{I}^\sigma$;
define ${S^{(\mu)}}_{ii} = P_\mu^t \, S^{(\mu)} \, P_\mu$, and ${S^{(\nu)}}_{ii} = P_\nu^t \, S^{(\nu)} \, P_\nu$;
compute ${S^{(\mu)}}_{ii}^{-1}$ through fast inversion;

`-- STEP 2.2 --`
compress $\tilde{S}^{(\nu)}_{ii} = {S^{(\nu)}}_{ii} - \hat{A}^{(\nu,\mu)} \, {S^{(\mu)}}_{ii}^{-1} \, \hat{A}^{(\mu,\nu)}$ as in Theorem 2.1;
compute ${\tilde{S}^{(\nu)}}_{ii}^{-1}$ through fast inversion;

`-- STEP 2.3 --`
compute $x_2 = {\tilde{S}^{(\nu)}}_{ii}^{-1}(z_2 - \hat{A}^{(\nu,\mu)} \, {S^{(\mu)}}_{ii}^{-1} \, z_1)$, and
$x_1 = {S^{(\mu)}}_{ii}^{-1} \, z_1 - {S^{(\mu)}}_{ii}^{-1} \, \hat{A}^{(\mu,\nu)} \, x_2$ through fast application;

Algorithm 1: Compression and fast application of ${\hat{A}^{(\sigma)}}_{ii}^{-1}$.

4.4. Approximate Matrix Factorization

Let A have dimension N and select a number of tree levels L so that each leaf box holds a sufficiently small number of dof's $m = N/2^L$ to allow for dense linear algebra manipulations at negligible cost. At the level of leaf-boxes, the Schur complements are computed and compressed to HBS-form in a straightforward manner at a cost proportional to $\mathrm{O}(m^3)$. Starting from the leaf-boxes, the Schur complements are merged together using the strategy described above, until the top tree-level is reached. The procedure is detailed in Algorithm 2. The cost of the algorithm is dominated by the cost of processing the boxes on the top level, namely $\mathrm{O}(\#\hat{B}^\sigma k^2)$, σ on level $\ell = 1$. Apart from the leaf nodes, the uncompressed Schur complements are never formed explicitly. Since the Gauss transforms L and M are obtained from the ${\hat{A}^{(\sigma)}}_{ii}^{-1}$ matrices, the fast merging process described in Algorithm 2 does, in fact, produce an approximate LDM^t factorization of A.

Since the Gauss transforms can be trivially inverted, the cost of inverting the LDM^t factorization is tantamount to the cost of inverting D which, in turn, see equation (11), coincides with the cost of inverting block $\hat{A}^{(0)}$, defined as:

$$\hat{A}^{(0)} = \begin{pmatrix} S^{(1)} & A^{(1,2)} \\ A^{(2,1)} & S^{(2)} \end{pmatrix}$$

input : A, binary tree partitioning the dof's of A
output: ${\hat{A}^{(\sigma)}_{ii}}^{-1}$, $S^{(\sigma)}$ for all σ's in the binary tree

`-- STEP 1 --`
for σ *on level* L **do**
  compute $S^{(\sigma)}$ by a straightforward approach;
  compress $S^{(\sigma)}$ by a straightforward approach;
end

`-- STEP 2 --`
for $\ell = L-1, \ldots, 1$ **do**
  for σ *on level* ℓ **do**
    form $\hat{A}^{(\sigma)}_{bb}$, $\hat{A}^{(\sigma)}_{bi}$, $\hat{A}^{(\sigma)}_{ii}$, $\hat{A}^{(\sigma)}_{ib}$;
    compress ${\hat{A}^{(\sigma)}_{ii}}^{-1}$ as in Algorithm 1;
    compress $S^{(\sigma)} = \hat{A}^{(\sigma)}_{bb} - \hat{A}^{(\sigma)}_{bi}\,{\hat{A}^{(\sigma)}_{ii}}^{-1}\,\hat{A}^{(\sigma)}_{ib}$ as in Theorem 2.1;
  end
end

Algorithm 2: Computation of Schur complements $S^{(\sigma)}$ through fast merging.

The inversion of $\hat{A}^{(0)}$ can be achieved with a slight modification of Algorithm 1, described in Algorithm 3. We conclude that the cost[8] of building the inverse factorization is

$$\mathrm{O}(\#\hat{B}^\sigma k^2) \quad = \quad \underset{LDM^t \text{ factorization}}{\mathrm{O}(\#\hat{B}^\sigma k^2)} \; + \; \underset{\text{inversion of } \hat{A}^{(0)}}{\mathrm{O}(\#\hat{B}^\sigma k^2)}$$

for σ on level $\ell = 1$.

In order to obtain cost estimates with respect to the problem size N, we evaluate the number of dof's in B^σ through a geometrical argument:

$$\#B^\sigma = \left(\frac{N}{2^\ell}\right)^{1-1/d} \quad , \quad \text{for box } \sigma \text{ on level } \ell \tag{13}$$

for some parameter d. The most conservative estimate is obtained for the limit case

[8] As previously, we assume $\#\hat{B}^\sigma = \mathrm{O}(\#\hat{I}^\sigma)$.

input : z_1, z_2, $S^{(1)}$, $S^{(2)}$, $\hat{A}^{(1,2)}$, $\hat{A}^{(2,1)}$
output: x_1, x_2, ${S^{(1)}}^{-1}$, ${\tilde{S}^{(2)}}^{-1}$

`-- STEP 1 --`
compute ${S^{(1)}}^{-1}$ through fast inversion;
compress $\tilde{S}^{(2)} = S^{(2)} - \hat{A}^{(2,1)}\, {S^{(1)}}^{-1}\, \hat{A}^{(1,2)}$ as in Theorem 2.1;
compute ${\tilde{S}^{(2)}_{ii}}^{-1}$ through fast inversion;

`-- STEP 2 --`
compute $x_2 = {\tilde{S}^{(2)}}^{-1}(z_2 - \hat{A}^{(2,1)}\, {S^{(1)}}^{-1}\, z_1)$, and $x_1 = {S^{(1)}}^{-1}\, z_1 - {S^{(1)}}^{-1}\, \hat{A}^{(1,2)}\, x_2$ through fast application;

Algorithm 3: Compression and fast application of ${\hat{A}^{(0)}}^{-1}$.

$d \uparrow \infty$, which yields:

$$\#B^\sigma = \frac{N}{2^\ell} \quad , \quad \text{for box } \sigma \text{ on level } \ell$$

Taking into account that $\#\hat{B}^\sigma \leq \#B^\sigma$, the cost of building the inverse factorization is

$$\mathrm{O}(\#\hat{B}^\sigma k^2) = \mathrm{O}(Nk^2) \quad , \quad \text{for box } \sigma \text{ on level } \ell = 1 \tag{14}$$

We conclude that the construction cost of the preconditioner, i.e., the inverse factorization, is linear with respect to N, provided that k is independent of N.

Let us briefly comment on the last statement. The assumption that k is independent of N, and solely dictated by the nature of the problem and the discretization, is unrealistic. In practice, we are interested in the asymptotic behavior of k, and some mild dependency upon N is acceptable, as long as $k/N \downarrow 0$. In this respect, estimate (14) is deceptively optimistic. On the other hand, taking $d \uparrow \infty$ in estimate (13) could be excessively conservative, and lead to an overly pessimistic result. As numerical experiments presented in Section 5 suggest, both of those scenarios are possible and the construction cost is tightly related to strategy used to increase the problem size. We defer further discussion to Section 5.

Finally, we remark that the inverse factorization $M^{-t}D^{-1}L^{-1}$ can be applied to a vector within linear complexity, thus making it suitable to be employed as a preconditioner.

5. Numerical Results

Discontinuous Galerkin (DG) finite element approximations provide a natural environment for testing the preconditioner. In fact, due to the doubling of dof's on the elements boundaries, the partitioning can be interpreted in purely geometrical terms, and the identification of interior and boundary dof's for each box is easily achieved. We investigate the behavior of the preconditioner for the following problems:

1. Poisson equation;
2. anisotropic reaction-diffusion equation;
3. Helmholtz equation;
4. Helmholtz equation with material contrast.

We rely upon Interior Penalty DG formulations and employ nodal DG finite elements discretiations, see [9]. All problems are formulated on the unit square $\Omega = (0,1)^2$, on which we lay a structured triangular grid. The domain is recursively partitioned as shown in Figure 4. We shall investigate the behavior of the preconditioner with respect to the partitioning scheme, the mesh size h, and the polynomial degree of approximation p. For the Helmholtz operator, the wave number κ is chosen as a function of the discretization parameters h and p, so that the dispersion error, see [1, 2], remains constant as the size of the problem is increased.

In order to assess the validity of our theoretical framework, we analyze the rank-structure of the Schur complements that emerge from discretizations of the Laplace and the Helmholtz operator. A qualitative study that compares the Schur complements of the two operators is illustrated in Figure 5. For a set problem size, we descend the tree focusing on the Schur complements relative to the first box on each level. At the top level, the Laplace operator exhibits a Schur complement whose off-diagonal blocks are rank-deficient, and more so than their counterparts for the Helmholtz operator. As we descend the tree, the difference between the Schur complements relative to the Laplace and Helmholtz operator becomes less evident. Further numerical experiments indicate that this behavior is consistent across problems of different sizes.

We proceed by studying the behavior of the HBS-rank k of the Schur complements relative to the Laplace operator as the problem size N is increased through h-refinements or p-enrichments, see Figure 6. The numerical experiments showed a uniformity of behavior of the Schur complements across all tree-levels for both operators. Thus, we limit our presentation to the Schur complement relative to the first

box on the top level. When h-refinements are performed, k is independent of the problem size while, in the case of p-enrichments, k grows roughly as the square root of the size n of the Schur complement. In the case of the Helmholtz operator, we postulate that k grows as $\log n$ in the case of h-refinements, and as $n^{1/2}$ in the case of p-enrichments.

Taking into account that estimate (13) reduces to $\#B^\sigma = \mathrm{O}(N^{1/2})$ in the case of h-refinements, and to $\#B^\sigma = \mathrm{O}(N)$ in the case of p-enrichments, we can revisit the cost estimate (14) as follows:

Laplace operator:

$$\text{cost of processing } \sigma \text{ on level } \ell \text{ } (h\text{-refinements}) = \frac{1}{2^\ell}\,\mathrm{O}\big(N^{1/2}\big) \tag{15a}$$

$$\text{cost of processing } \sigma \text{ on level } \ell \text{ } (p\text{-enrichments}) = \frac{1}{2^\ell}\,\mathrm{O}\big(N^{2}\big) \tag{15b}$$

Helmholtz operator:

$$\text{cost of processing } \sigma \text{ on level } \ell \text{ } (h\text{-refinements}) = \frac{1}{2^\ell}\,\mathrm{O}\big(N^{1/2}\log^2 N\big) \tag{16a}$$

$$\text{cost of processing } \sigma \text{ on level } \ell \text{ } (p\text{-enrichments}) = \frac{1}{2^\ell}\,\mathrm{O}\big(N^{2}\big) \tag{16b}$$

The estimates corresponding to the two scenarios, i.e., h-refinements as opposed to p-enrichments, are dramatically different. Apart from the different growth rates of k, this is dictated by the following geometrical argument. Given a fixed box, e.g., a box on the top tree-level, when h-refinements are performed, we observe a "thinning" of the boundary, namely the number of boundary dof's grows slower than the total number of dof's in the box. On the other hand, in the case of p-enrichments which corresponds to the limit $d \uparrow \infty$, the box contains a fixed number of elements, and no "thinning" of the boundary occurs. In applications of practical interest, the problem size is increased through adaptive strategies, that combine h-refinements and p-enrichments to produce optimal meshes. Estimates (15b) and (16b) should be regarded as the worst-case scenarios, which are not to be encountered in practice.

Although "time" (construction and application) is the ultimate measure of performance of a preconditioner, it is spectacularly implementation-dependent, and it makes little sense in the context of a non-optimized implementation. We take the

Figure 4: Domain partitions.

following approach. Since the Schur complements are compressed by recursively performing ID's of appropriately select sub-blocks, we estimate the total compression cost through (3). More precisely, we select the anticipated rank l according to the above discussion, and employ l and the actual rank k, as returned by the ID, in the computation of the final estimate. We investigate the agreement between such empirical cost and the theoretical estimates (15) and (16). We measure the performance of the preconditioner in terms of GMRES iterations. If l is properly selected, as a function of N, we expect the performance of the preconditioner not to degrade as the problem size is increased. We compare the performance of our preconditioner to that of a standard ILU preconditioner.

As a first example, we consider the Poisson equation on Ω, with a homogenous Dirichlet boundary condition:

$$\begin{aligned} -\Delta u &= f \qquad \text{in } \Omega \\ u &= 0 \qquad \text{on } \partial\Omega \end{aligned}$$

As anticipated, the problem is discretized using an Interior Penalty DG formulation. We study the behavior of the preconditioner and compare it to that of a standard ILU preconditioner. The construction cost agrees with the theoretical estimates (15), and the performance, i.e., the number of GMRES iterations, is independent of h, and p, see Figure 7.

Finally, we move to hyperbolic problems. Let us consider the Helmoltz equation on Ω with a homogenous Dirichlet boundary condition:

$$\begin{aligned} -\Delta u - \kappa^2 u &= f \qquad \text{in } \Omega \\ u &= 0 \qquad \text{on } \partial\Omega \end{aligned}$$

This boundary value problem describes propagation of forced waves inside a soft cavity. Although exterior scattering problems are often of greater interest, the cavity problem is computationally more challenging because of possible resonances. Remarkably, the behavior of the preconditioner is qualitatively similar to that observed in the case of the Poisson problem. The construction cost is in agreement with estimates (16) and the performance of the preconditioner does not deteriorate as the problem size is increased, see Figure 8.

6. Conclusions

We have presented the construction of a preconditioner that exploits low-rank compression of Schur complements. The construction can be viewed as a variant of the well-known nested dissection algorithm, since it employs a reordering of the degrees of freedom which allows for an advantageous elimination order. Our approach, originally inspired by the work of [8], follows a black-box approach that gives the construction the flexibility to be applied to a number of discretization techniques, such as finite differences, CG finite elements and DG finite elements. The preconditioner can be applied within linear complexity, and we provide an estimate of the construction cost. Such estimate depends widely upon the strategy used to vary the problem size (h-refinement as opposed to p-enrichments, in the case of finite elements approximations), with a worst-case-scenario of quadratic growth. Although this issue requires further investigation, we believe that for applications of practical interest, the construction cost is within linear growth. We tested the performance of the preconditioner on DG approximations of elliptic as well as hyperbolic problems, see [9], and demonstrated its robustness. More specifically, the preconditioned system is solved within a number of GMRES iterations that is independent of both the mesh size and the order of approximation. The choice of DG finite elements approximations was dictated by implementation convenience, namely the reordering of the degrees of freedom has a straightforward geometrical interpretation, and should not be viewed as a limitation of the applicability of the preconditioner. In principle, the reordering of the degrees of freedom can be performed with any graph-partitioning software.

The construction of preconditioners for linear systems that arise from wave propagation phenomena is notoriously challenging, see [11]. The proposed preconditioner is based on a completely general construction and has proven effective for a number of problems. In terms of future research developments, the most pressing issue is to verify the agreement between the actual construction cost and its theoretical estimate through an optimized implementation. This will allow us to assess whether the asymptotic region is reached for problem sizes of practical relevance. Finally, we should investigate the effectiveness of the preconditioner for a larger problems, e.g., coupled multi-physics problems.

References

[1] M. Ainsworth. Dispersive and dissipative behaviour of high order discontinuous galerkin finite element methods. *Journal of Computational Physics*, 198(1):106–130, 2004.

[2] M. Ainsworth, P. Monk, and W. Muniz. Dispersive and dissipative properties of discontinuous galerkin finite element methods for the second-order wave equation. *Journal of Scientific Computing*, 27(1-3):5–40, 2006.

[3] M.P. Bendsøe and O. Sigmund. *Topology Optimization*. World Scientific, 2009.

[4] S. Börm. Approximation of solution operators of elliptic partial differential equations by $\mathcal{H}$- and $\mathcal{H}^2$-matrices. *Numerische Mathematik*, 115(2):165–193, 2010.

[5] T.F. Chan and P.C. Hansen. Some applications of the rank revealing QR factorization. *SIAM Journal on Scientific and Statistical Computing*, 13(3):727–741, 1992.

[6] A. George. Nested dissection of a regular finite element mesh. *SIAM Journal on Numerical Analysis*, 10(2):345–363, 1973.

[7] A. Gillman. *Fast direct solvers for elliptic partial differential equations*. PhD thesis, University of Colorado, Boulder, 2011.

[8] A. Gillman and P.G. Martinsson. An O(n) algorithm for constructing the solution operator to 2d elliptic boundary value problems in the absence of body loads. *Advances in Computational Mathematics*, 40(4):773–796, 2014.

[9] J.S. Hesthaven and T. Warburton. *Nodal discontinuous Galerkin methods: algorithms, analysis, and applications*. Springer Science & Business Media, 2007.

[10] P.G. Martinsson. A fast randomized algorithm for computing a hierarchically semiseparable representation of a matrix. *SIAM J. Matrix Anal. Appl.*, 32(4):1251–1274, November 2011.

[11] A. Toselli and O. Widlund. *Domain decomposition methods: algorithms and theory*, volume 3. Springer, 2005.

[12] F. Woolfe, E. Liberty, V. Rokhlin, and M. Tygert. A fast randomized algorithm for the approximation of matrices. *Appl. Comput. Harmon. Anal.*, 25(3):335–366, November 2008.

[13] J. Xia, S. Chandrasekaran, M. Gu, and X.S. Li. Superfast multifrontal method for large structured linear systems of equations. *SIAM Journal on Matrix Analysis and Applications*, 31(3):1382–1411, 2009.

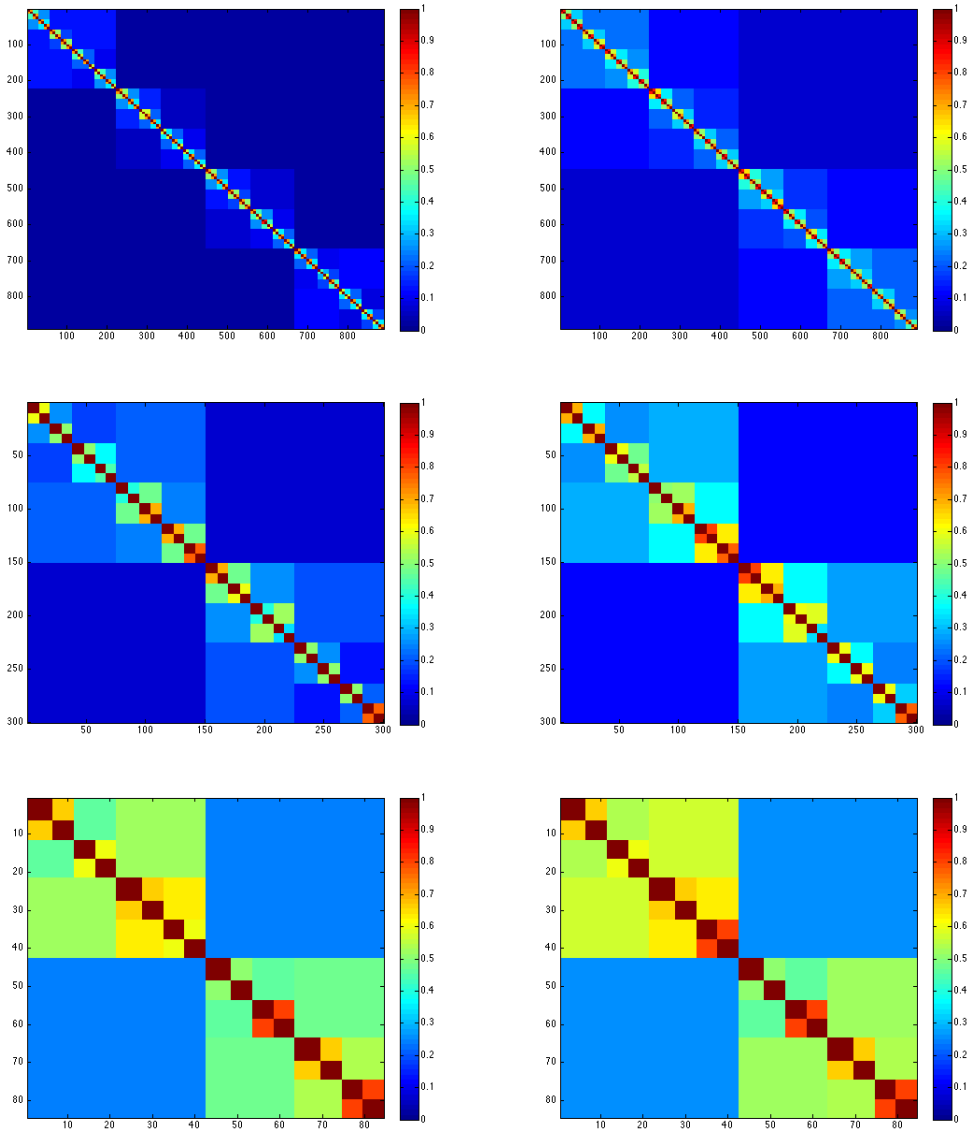


Figure 5: rank-structure of Schur complements arising from the discretization of the Laplace (left column) and Helmholtz (right column) operators. From top to bottom, finer tree-levels are considered. The colors indicate the relative rank of the corresponding sub-blocks.

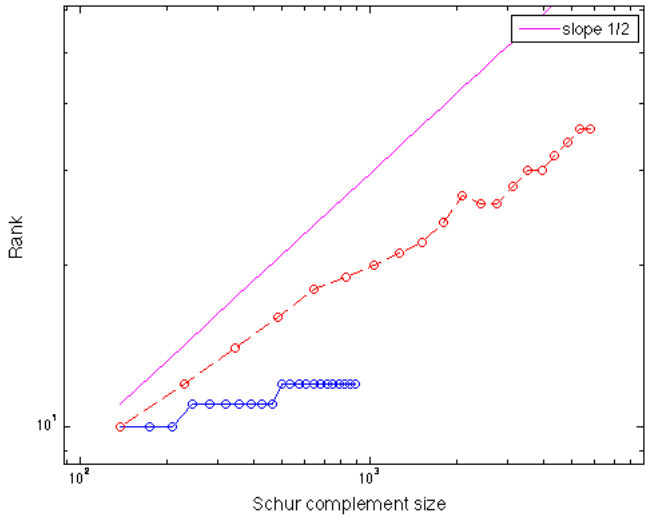
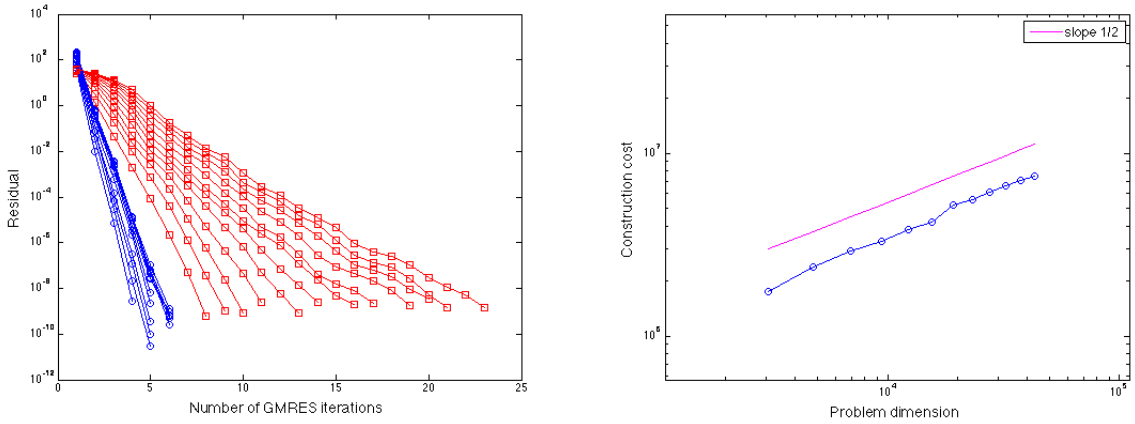
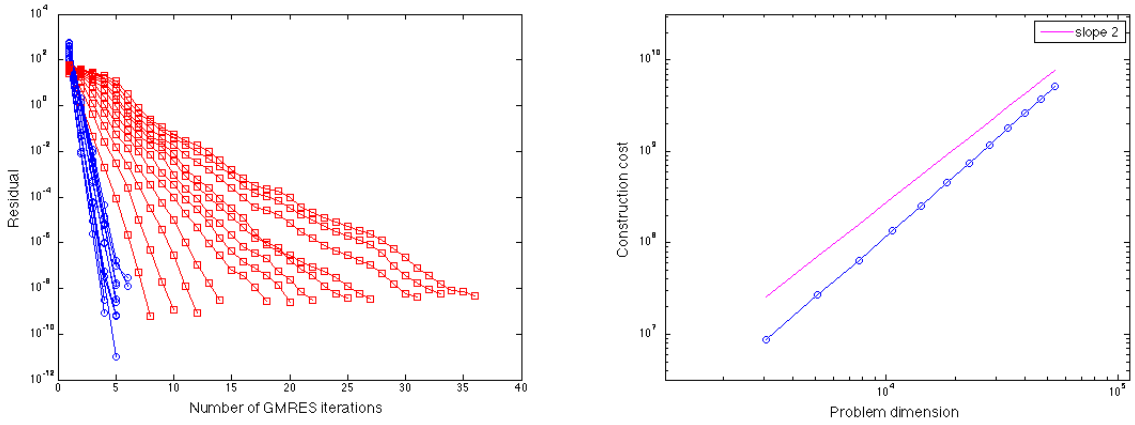


Figure 6: Rank growth of Schur complements. Data refer to ranks of largest off-diagonal block of top level Schur complements for the Laplace operator. The problem size is increased through h-refinements (solid blue line) or p-enrichments (dashed red line.)

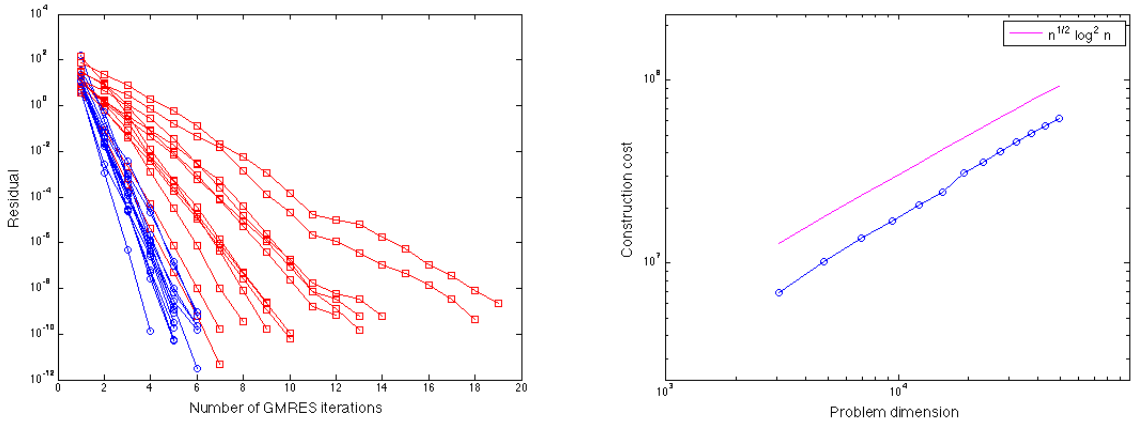


(a) Problem size is increased through h-refinements

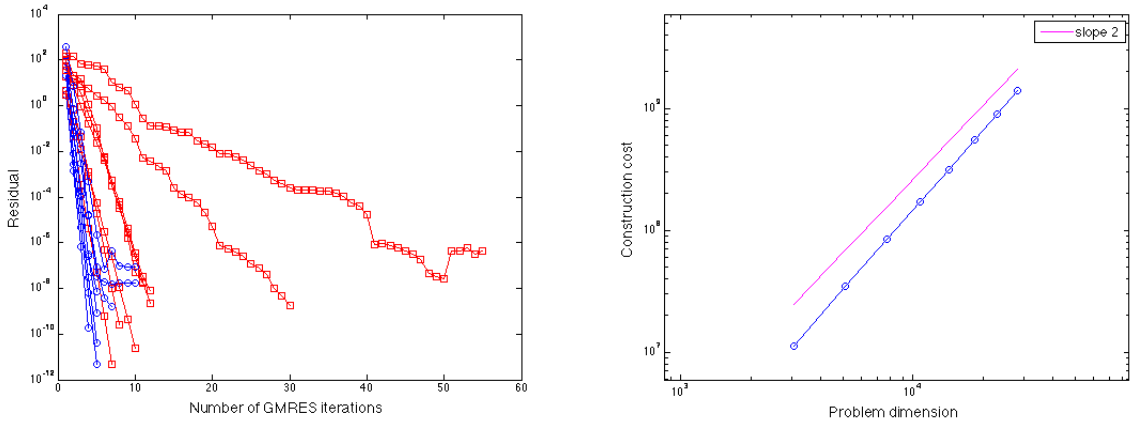


(b) Problem size is increased through p-enrichments

Figure 7: comparison between the proposed preconditioner and a standard ILU preconditioner, for the solution of a Poisson problem through GMRES. Squares refer to ILU, while circles refer to the proposed preconditioner. Unlike ILU, the performance, i.e., the number of GMRES iterations, does not deteriorate as the problem size increases. The theoretical construction cost is in accordance with estimates (15).



(a) Problem size is increased through h-refinements



(b) Problem size is increased through p-enrichments

Figure 8: comparison between the proposed preconditioner and a standard ILU preconditioner, for the solution of a Helmholtz problem through GMRES. Squares refer to ILU, while circles refer to the proposed preconditioner. Unlike ILU, the performance, i.e., the number of GMRES iterations, does not deteriorate as the problem size increases. The theoretical construction cost is in accordance with estimates (15).