Efficient preconditioning of hp-FEM matrices arising from time-varying problems: an application to topology optimization

P. Gatto[a,*], J.S. Hesthaven[b], R.E. Christiansen[c]

[a] *MATHICSE, École Polytechnique Fédérale de Lausanne (EPFL), MA C2 585 (Bâtiment MA), Station 8, CH-1015 Lausanne, Switzerland*

[b] *MATHICSE, École Polytechnique Fédérale de Lausanne (EPFL), MA C2 652 (Bâtiment MA), Station 8, CH-1015 Lausanne, Switzerland*

[c] *Technical University of Denmark, Nils Koppels Allé, Building 404, room 131, 2800 Kgs. Lyngby, Denmark*

Abstract

We previously introduced a preconditioner that has proven effective for hp-FEM discretizations of various challenging elliptic and hyperbolic problems. The construction is inspired by standard nested dissection, and relies on the assumption that the Schur complements can be approximated, to high precision, by Hierarchically-Semi-Separable matrices. The preconditioner is built as an approximate LDM^t factorization through a divide-and-conquer approach. This implies an enhanced flexibility which allows to handle unstructured geometric meshes, anisotropies, and discontinuities. We build on our previous numerical experiments and develop a preconditioner-update strategy that allows us handle time-varying problems. We investigate the performance of the precondition along with the update strategy in context of topology optimization of an acoustic cavity.

Keywords: Preconditioned GMRES, Interpolative Decomposition, Indefinite operators, Time-varying problems, Acoustic Topology Optimization

[*] Corresponding author

Email addresses: `paolo.gatto@epfl.ch` (P. Gatto), `jan.hesthaven@epfl.ch` (J.S. Hesthaven), `raelch@mek.dtu.dk` (R.E. Christiansen)

1. Introduction

In this work, we apply the construction introduced in [11] to discretizations of problems with strong material discontinuities and time-varying coefficients. The core of the construction rests on the observation that, for a large class of problems, the dense Schur complement matrices that arise in the nested dissection method display rank-deficient off-diagonal blocks. In the case of well-behaved elliptic problems, this property can be traced to the separability of the underlying Green's function, see, e.g., [5, 27]. To the contrary, this argument does not apply to wave-propagation problems, and, in the high-frequency limit, this property indeed ceases to hold, see [9]. Nevertheless, in the case of moderate frequencies, the off-diagonal blocks of the Schur complements can, for all practical purposes, be treated as low-rank. The intuitive explanation of this phenomenon is that, when considered on sufficiently small subdomains, the solution of wave problems resemble that of elliptic problems. By exploiting this property, approximate matrix decompositions can be constructed cheaply, and turn out to be excellent preconditioners.

A number of compressed-rank formats has appeared in the literature. Hierarchical matrices, or $\mathcal{H}$-matrices, were first defined in the seminal work of Hackbusch, see [17]. Subsequently, the subclass of $\mathcal{H}^2$-matrices was introduced in [18]. Those matrices are attractive because allow traditionally expensive operations to be carried out within linear complexity, see, e.g., [2] for a detailed discussion. Hierarchically Semi-Separable (HSS) matrices were proposed by Chandrasekaran et al. in [6], and are closely related to $\mathcal{H}$- and $\mathcal{H}^2$-matrices. Fast algorithms for their manipulation have been proposed, among others, by Sheng et al. [22], Xia et al. [28], Martinsson [20], and Gillman and Martinsson [14].

As described in details in [11], the preconditioner construction combines a classical method, i.e., an LDM^t factorization, with an approximation scheme for the resulting dense Schur complements. In fact, this approach is well established in the literature, see, e.g., Grasedyck et al. [15], and Xia et al. [26]. While our construction was largely inspired by the work of Gillman and Martinsson done in the context of finite difference discretizations, see [13], its main novelty is that we exploited the geometric flexibility provided by the LDM^t factorization to accommodate for finite elements discretizations and unstructured meshes. In fact, while the main body of the literature deals with finite difference approximations, in the context of finite element discretizations we are only aware of the work by Aminfar and Darve, see [1].

In this work we describe the application of the preconditioner to high-contrast, time-varying Helmoltz problems arising in the context of acoustic Topology Optimization, see [4]. Topology Optimization is an iterative method that creates highly optimized designs by determining a distribution of material that fulfills a specific task. Typically, the method requires 100 through 1000 iterations to recover a locally optimal and physically admissible design. Consequently, the governing equations for the problem under consideration must be solved a large number of times for a slowly changing material distribution. When systems of several millions of degrees of freedom are considered, as is often the case for real world applications, their solutions through traditional direct techniques becomes infeasible. Naturally, this raises the interest in using highly scalable parallel iterative techniques. For physical problems governed by the Helmholtz equation, like acoustic, electromagnetic and structural vibration problems, no general scalable parallel iterative techniques currently exist. Therefore it is of interest to investigate the performance of the preconditioner in the context of Topology Optimization.

The paper is organized as follows. In Section 2 we present an improved analytical apparatus that allows us to better characterize the Schur complements as solution operators. In Section 3 we recall the construction of the preconditioner in order to make this work self-contained. Section 4 is devoted to numerical results. Finally, in Section 5, we draw conclusions from this work, and point towards future directions of research.

2. Analytical Apparatus

We provide an insight into the rank-structure of the Schur complements that arise in the construction of the preconditioner. Let A be a finite element discretization of a boundary value problem posed on a domain Ω. Since A is sparse, we can reorder its degrees of freedom (dof's) to expose the following block-structure, and define the (aggregated) sub-matrices $A^{(k)}$:

$$A = \begin{pmatrix} {A^{(1)}}_{ii} & & {A^{(1)}}_{ib} & \\ & {A^{(2)}}_{ii} & & {A^{(2)}}_{ib} \\ {A^{(1)}}_{bi} & & {A^{(1)}}_{bb} & A^{(1,2)} \\ & {A^{(2)}}_{bi} & A^{(2,1)} & {A^{(2)}}_{bb} \end{pmatrix} \quad , \quad A^{(k)} = \begin{pmatrix} {A^{(k)}}_{ii} & {A^{(k)}}_{ib} \\ {A^{(k)}}_{bi} & {A^{(k)}}_{bb} \end{pmatrix} \quad k = 1, 2$$

The diagonal blocks in the partitions are square sub-matrices. Since each dof j is associated to a unique finite element basis function φ_j with localized support, we define the following subdomains of Ω:

$$\overline{\Omega^{(k)}}_i = \cup \{\operatorname{supp} \varphi_j \ : \ j \in \operatorname{ind}({A^{(k)}}_{ii})\} \quad , \quad \overline{\Omega^{(k)}}_b = \cup \{\operatorname{supp} \varphi_j \ : \ j \in \operatorname{ind}({A^{(k)}}_{bb})\}$$

along with $\overline{\Omega^{(k)}} = \overline{\Omega^{(k)}{}_i} \cup \overline{\Omega^{(k)}{}_b}$. Here $\mathrm{ind}(A^{(k)}{}_{ii})$ and $\mathrm{ind}(A^{(k)}{}_{bb})$ are the row or, equivalently, column indices of A that form blocks $A^{(k)}{}_{ii}$ and $A^{(k)}{}_{bb}$, respectively. A geometrical interpretation of this construction is illustrated in Figure 1. In vague terms, we can think of $\Omega^{(k)}{}_i$ as the portion of subdomain $\Omega^{(k)}$ "well contained" in its interior, and of $\Omega^{(k)}{}_b$ as the portion adjacent to the boundary $\partial\Omega^{(k)}$. Hence the choice of subscripts in the matrix partitioning.

An LDM^t factorization of A is immediately obtained as:

$$A = L \underset{D}{\left(\begin{array}{cc|cc} A^{(1)}{}_{ii} & & & \\ & A^{(2)}{}_{ii} & & \\ \hline & & S^{(1)} & A^{(1,2)} \\ & & A^{(2,1)} & S^{(2)} \end{array}\right)} M^t$$

for suitable factors L and M. The Schur complements $S^{(k)}$ are defined as:

$$S^{(k)} = A^{(k)}{}_{bb} - A^{(k)}{}_{bi}\, {A^{(k)}{}_{ii}}^{-1}\, A^{(k)}{}_{ib}$$

When we restrict the original boundary value problem to $\Omega^{(k)}$, complement it with a homogenous Dirichlet boundary condition on $\mathrm{int}(\partial\Omega^{(k)} \setminus \Gamma)$, and approximate it in the same fashion as the original problem, we obtain the discrete operator $A^{(k)}$. A block-solve yields

$$u^{(k)}{}_b = {S^{(k)}}^{-1} \big(f^{(k)}{}_b - A^{(k)}{}_{bi}\, {A^{(k)}{}_{ii}}^{-1}\, f^{(k)}{}_i \big)$$

where $u^{(k)}{}_b$ is the restriction of the solution vector to the dof's contained in $\Omega^{(k)}{}_b$, and, similarly, the load vector has been partitioned as $(f^{(k)}{}_i, f^{(k)}{}_b)$. Thus, ${S^{(k)}}^{-1}$ is the discrete analog of the solution operator of the reduced problem on $\Omega^{(k)}$, restricted to the dof's in $\Omega^{(k)}{}_b$.

The factorization proceeds by exploiting the fact that, after a suitable permutation, the bottom-right super-block of D exhibits the same structure as A. To this end, we select some dof's of $S^{(1)}$ and $S^{(2)}$ for elimination, and refer to the associated subdomain as $\hat{\Omega}_i$. We remark that this selection process can be understood under either a purely algebraic, or geometric point of view. In the the latter case, this is equivalent to identify new interior dof's among those pertaining to $\Omega^{(1)}{}_b$ or $\Omega^{(2)}{}_b$, see

Figure 2. Up to a permutation, we can repartition the Schur complements as:

$$S^{(k)} \xrightarrow{\text{repartition}} \begin{pmatrix} {S^{(k)}}_{ii} & {S^{(k)}}_{ib} \\ {S^{(k)}}_{bi} & {S^{(k)}}_{bb} \end{pmatrix}$$

where $\operatorname{ind}({S^{(k)}}_{ii})$ are the indices of the dof's in ${\Omega^{(k)}}_b$ also associated to $\hat{\Omega}_i$, and $\operatorname{ind}({S^{(k)}}_{bb})$ are all the other indices in ${\Omega^{(k)}}_b$. Naturally, the indices of the off-diagonal blocks can be inferred from those of the adjacent diagonal blocks. We manipulate the bottom-right super block of D as:

$$\begin{pmatrix} S^{(1)} & A^{(1,2)} \\ A^{(2,1)} & S^{(2)} \end{pmatrix} \xrightarrow{\text{repartition}} \left(\begin{array}{cc|cc} {S^{(1)}}_{ii} & {S^{(1)}}_{ib} & {A^{(1,2)}}_{ii} & {A^{(1,2)}}_{ib} \\ {S^{(1)}}_{bi} & {S^{(1)}}_{bb} & {A^{(1,2)}}_{bi} & {A^{(1,2)}}_{bb} \\ \hline {A^{(2,1)}}_{ii} & {A^{(2,1)}}_{ib} & {S^{(2)}}_{ii} & {S^{(2)}}_{ib} \\ {A^{(2,1)}}_{bi} & {A^{(2,1)}}_{bb} & {S^{(2)}}_{bi} & {S^{(2)}}_{bb} \end{array}\right)$$

$$\xrightarrow{\text{permute}} \left(\begin{array}{cc|cc} {S^{(1)}}_{ii} & {A^{(1,2)}}_{ii} & {S^{(1)}}_{ib} & {A^{(1,2)}}_{ib} \\ {A^{(2,1)}}_{ii} & {S^{(2)}}_{ii} & {A^{(2,1)}}_{ib} & {S^{(2)}}_{ib} \\ \hline {S^{(1)}}_{bi} & {A^{(1,2)}}_{bi} & {S^{(1)}}_{bb} & {A^{(1,2)}}_{bb} \\ {A^{(2,1)}}_{bi} & {S^{(2)}}_{bi} & {A^{(2,1)}}_{bb} & {S^{(2)}}_{bb} \end{array}\right) \xrightarrow{\text{regroup}} \begin{pmatrix} {\hat{A}^{(0)}}_{ii} & \hat{A}^{(1,2)} \\ \hat{A}^{(2,1)} & {\hat{A}^{(0)}}_{bb} \end{pmatrix}$$

Finally, we obtain the factorization:

$$A = L \begin{pmatrix} {A^{(1)}}_{ii} & & & \\ & {A^{(2)}}_{ii} & & \\ & & {\hat{A}^{(0)}}_{ii} & \\ & & & S^{(0)} \end{pmatrix} M^t$$

where L and M are new suitable accumulated Gauss transforms, and the Schur complement $S^{(0)}$ is defined as:

$$S^{(0)} = {\hat{A}^{(0)}}_{bb} - \hat{A}^{(2,1)} \, {{\hat{A}^{(0)}}_{ii}}^{-1} \, \hat{A}^{(1,2)}$$

Because of the structure of the factors L and M, when we invert the factorization, it is easy to see that the bottom-right block of A^{-1} coincides with ${S^{(0)}}^{-1}$. Thus, if we define $\overline{\Omega_b} = \cup\{\operatorname{supp}\varphi_j \;:\; j \in \operatorname{ind}({\hat{A}^{(0)}}_{bb})\}$, see Figure 2, we can characterize ${S^{(0)}}^{-1}$ as the restriction to Ω_b of the solution operator of the original problem posed on Ω.

In the case of discretizations of elliptic boundary value problems, it is well-established that the solution operator exhibits low-rank long-range interactions, see, e.g., [3], [17], and [6]. On the other hand, in the case of a wave-propagation Helmholtz problem, the underlying Green's function exhibits low-rank long-range interactions only for low frequencies, see [9] for an extensive discussion. Nevertheless, in special cases, a directional low-rank property holds, and the separability of the Green's function can still be attained, see [21], [10]. We only provide a heuristic explanation of the rather encouraging numerical results attained by our construction. Anticipating the discussion in Section 3, let us suppose that we have reparationed subdomains $\Omega^{(k)}$ into smaller subdomains, and so on. As the subdomains become smaller and smaller in terms of wavelengths, the ranks of the solution operator of the Helmholtz problem approach those of the Laplace problem. At the same time, when we consider larger and lager subdomains, the boundary domains $\Omega_b^{(k)}$ increasingly approach elongated structures, which enjoy separability of the associated Green's function.

3. Preconditioner Construction and Update

The construction relies on a variant of the well-known nested dissection algorithm, see George [12], and extends the work of Gilmann and Martinsson, see [13], developed in the context of finite difference approximations.

3.1. Matrix Reordering

Let A be a sparse matrix of dimension N arising from the discretization of a differential operator, and partition its dof's into two boxes, $\sigma_1 = I^{\sigma_1} \cup B^{\sigma_1}$ and $\sigma_2 = I^{\sigma_2} \cup B^{\sigma_2}$, so that the following connectivity graph holds:

$$\begin{tikzcd}
B^{\sigma_1} \arrow[r, leftrightarrow] \arrow[d, leftrightarrow] & B^{\sigma_2} \arrow[d, leftrightarrow] \\
I^{\sigma_1} & I^{\sigma_2}
\end{tikzcd} \tag{1}$$

The boxes are connected to each other through their boundaries dof's B^1 and B^2, while the interior dof's I^1 and I^2 do not interact with each other. We repartition each box into a pair of sibling boxes, in fact creating a binary tree of boxes, see Figure 3, and identify boundary and interior dof's, in the sense of graph (1), for each new pair of sibling boxes. For illustration, the connectivity graph of boxes on the second tree level, i.e., $\ell = 2$, is:

$$\begin{tikzcd}
B^{\sigma_3} \arrow[r, leftrightarrow] \arrow[d, leftrightarrow] \arrow[rr, bend left, leftrightarrow] \arrow[rrr, bend left, leftrightarrow] & B^{\sigma_4} \arrow[r, leftrightarrow] \arrow[d, leftrightarrow] \arrow[rr, bend left, leftrightarrow] & B^{\sigma_5} \arrow[r, leftrightarrow] \arrow[d, leftrightarrow] & B^{\sigma_6} \arrow[d, leftrightarrow] \\
I^{\sigma_3} & I^{\sigma_4} & I^{\sigma_5} & I^{\sigma_6}
\end{tikzcd}$$

The construction terminates at tree level $\ell = L$, when the newly created boxes, which we shall refer to as "leaf-boxes," contain a number of dof's m such that $2^L \, \mathrm{O}(m^3) = \mathrm{O}(N)$. When this is the case, we say that the leaf-boxes are sufficiently small to allow for dense linear algebra operations at a negligible cost.

We establish an ordering for the dof's I^σ and B^σ of each box 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)

Here the MATLAB®-like notation $A(I, J)$ indicates the restriction of matrix A to row-index vector I and column-index vector J. Let $n = 2 + 2^2 + \cdots + 2^L = 2^{L+1} - 2$ be the total number of boxes, and let n_ℓ be the index of the first box belonging to tree level ℓ, with the convention that $n_{L+1} - 1 = n$. Henceforth, when no confusion arises, to simplify notation, we shall identify a box to its index. Thus, $\{n, \ldots, n_L\}$ are the leaf boxes, listed in reversed order. We order the dof's by grouping together the interior dof's $\{I^n, \ldots, I^{n_L}\}$ and the boundary dof's $\{B^n, \ldots, B^{n_L}\}$, so that the following block-structure emerges:

$$A = \left(\begin{array}{ccc|ccc} {A^{(n)}}_{ii} & & & {A^{(n)}}_{ib} & & \\ & \ddots & & & \ddots & \\ & & {A^{(n_L)}}_{ii} & & & {A^{(n_L)}}_{ib} \\ \hline {A^{(n)}}_{bi} & & & {A^{(n)}}_{bb} & \cdots & A^{(n,n_L)} \\ & \ddots & & \vdots & \ddots & \vdots \\ & & {A^{(n_L)}}_{bi} & A^{(n_L,n)} & \cdots & {A^{(n_L)}}_{bb} \end{array}\right) \tag{2}$$

Here, as common practice, we have omitted the null blocks.

Ideally, grid information, in conjunction with information about the nature of the differential operator, should be employed to produce a competitive ordering of the dof's. The advantage of describing the construction in algebraic, rather than geometric, terms is twofold: on one hand it allows to handle virtually all types of discretization techniques (finite differences, CG and DG finite elements, high order elements, high smoothness IGA discretizations, etc.); on the other, when no grid information is available, it allows for the use of a graph partitioner, e.g., METIS, see [19].

3.2. Approximate Factorization

In a nutshell, the factorization is achieved by sweeping over all boxes, starting from the leaf-boxes, and decoupling the interior dof's from the boundary dof's through Gauss transforms. For each box σ, we define $L^{(\sigma)}$ and $M^{(\sigma)}$ as the unit-lower-triangular matrices such that

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

They are called, respectively, the left and right Gauss transform associated to σ. Here we set ${\hat{A}^{(\sigma)}}_{ii} = {A^{(\sigma)}}_{ii}$ for the leaf-boxes, while the definition of ${\hat{A}^{(\sigma)}}_{ii}$ for non-leaf boxes is given in equation (4) and there discussed. For each tree level ℓ, we define the accumulated Gauss transforms

$$L_{(\ell)} = L^{(n_{\ell+1}-1)} \cdots L^{(n_\ell)} \qquad , \qquad M_{(\ell)} = M^{(n_{\ell+1}-1)} \cdots M^{(n_\ell)}$$

and employ $L_{(L)}$ and $M_{(L)}$ to decouple the top-left super-block of A:

$$L_{(L)}{}^{-1}\,A\,M_{(L)}{}^{-t} =$$
$$\left(\begin{array}{ccc|cccc} A^{(n)}{}_{ii} & & & & & & \\ & \ddots & & & & & \\ & & A^{(n_L)}{}_{ii} & & & & \\ \hline & & & S^{(n)} & A^{(n,n-1)} & \cdots & A^{(n,n_L)} \\ & & & A^{(n-1,n)} & \ddots & \ddots & \vdots \\ & & & \vdots & \ddots & \ddots & A^{(n_L+1,n_L)} \\ & & & A^{(n_L,n)} & \cdots & A^{(n_L,n_L+1)} & S^{(n_L)} \end{array}\right) \tag{3}$$

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.

We exploit the box hierarchy to recursively proceed in the factorization. For each pair of sibling boxes $\{\mu,\nu\}$ with common ancestor σ, we define

$$\hat{I}^{\sigma} = (B^{\mu} \cup B^{\nu}) \cap I^{\sigma} \qquad , \qquad \hat{B}^{\sigma} = (B^{\mu} \cup B^{\nu}) \cap B^{\sigma}$$

which can be interpreted, respectively, as the residual interior and boundary dof's. Note that $\{\hat{I}^{\sigma}, \hat{B}^{\sigma}\}$ 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 of B^{μ} and $\{B^{\nu} \cap \hat{I}^{\sigma}, B^{\nu} \cap \hat{B}^{\sigma}\}$ is a partition of B^{ν}. Consequently, up to a permutation, we can 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:

$$\hat{A}^{(\sigma)}{}_{ii} = \begin{pmatrix} S^{(\mu)}{}_{ii} & \hat{A}^{(\mu,\nu)} \\ \hat{A}^{(\nu,\mu)} & S^{(\nu)}{}_{ii} \end{pmatrix} \; ; \; \hat{A}^{(\sigma)}{}_{ib} = \begin{pmatrix} S^{(\mu)}{}_{ib} & \hat{A}^{(\mu,\nu)} \\ \hat{A}^{(\nu,\mu)} & S^{(\nu)}{}_{ib} \end{pmatrix} \; ; \; \hat{A}^{(\sigma)}{}_{bb} = \begin{pmatrix} S^{(\mu)}{}_{bb} & \hat{A}^{(\mu,\nu)} \\ \hat{A}^{(\nu,\mu)} & S^{(\nu)}{}_{bb} \end{pmatrix} \tag{4}$$

and matrix $\hat{A}^{(\sigma)}{}_{bi}$ analogously to $\hat{A}^{(\sigma)}{}_{ib}$. In order to simplify the notation, $\hat{A}^{(\mu,\nu)}$ indicates a generic sub-matrix of $A^{(\mu,\nu)}$ whose row and column index vectors can be inferred from the context, namely the adjacent Schur complement blocks. Thus, distinct instances of the symbol should be regarded *a priori* as different matrices. We reorder the dof's $\{B^n, \ldots, B^{n_L}\}$ of the bottom-right super-block of ${L_{(L)}}^{-1} A\, {M_{(L)}}^{-t}$ as $\{\hat{I}^{n_L-1}, \ldots, \hat{I}^{n_{L-1}}, \hat{B}^{n_L-1}, \ldots, \hat{B}^{n_{L-1}}\}$. In practice, this corresponds to apply a permutation to the rows and columns of ${L_{(L)}}^{-1} A\, {M_{(L)}}^{-t}$. However, since it is well understood how Gauss transforms and permutations commute with each other, we systematically drop to latter ones from the exposition. Thus, we can rewrite equation (3) as:

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

where the symbol $\star$ indicates an omitted non-zero block.

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. It exploits the fact that the structure of the bottom-right super-block is identical to that of the original matrix A, cf. equation (2). 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. We shall refer to this action as “processing the box.” Its most delicate step is the fast inversion of $\hat{A}^{(\sigma)}{}_{ii}$ through the block-solver described in Algorithm 1. The crucial point, see [11], is that, once the Schur complements of the child-boxes are in HSS form, then the action of the Schur complement of the parent box can be determined within linear complexity. This is a sufficient condition to compress the Schur complement to HSS form within linear complexity as well, see [20].

```
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 1 --
determine permutations $P_\mu : B^\mu \rightarrow B^\mu \cap \hat{I}^\sigma$, and $P_\nu : B^\nu \rightarrow B^\nu \cap \hat{I}^\sigma$;
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 --
compress ${\tilde{S}^{(\nu)}}_{ii} = {S^{(\nu)}}_{ii} - \hat{A}^{(\nu,\mu)}\, {S^{(\mu)}}_{ii}^{-1}\, \hat{A}^{(\mu,\nu)}$ to HSS form;
compute ${\tilde{S}^{(\nu)}}_{ii}^{-1}$ through fast inversion;
-- STEP 3 --
compute $x_2 = {\tilde{S}^{(\nu)}}_{ii}^{-1}(z_2 - \hat{A}^{(\nu,\mu)}\, {S^{(\mu)}}_{ii}^{-1}\, z_1)$ through fast application;
compute $x_1 = {S^{(\mu)}}_{ii}^{-1}\, z_1 - {S^{(\mu)}}_{ii}^{-1}\, \hat{A}^{(\mu,\nu)}\, x_2$ through fast application;
```

Algorithm 1: Construction of a fast block-solver for ${\hat{A}^{(\sigma)}}_{ii}$.

When the decoupling process terminates, we obtain the following factorization:

$$L^{-1} A M^{-t} = \begin{pmatrix} {\hat{A}^{(n)}}_{ii} & & & & \\ & \ddots & & & \\ & & {\hat{A}^{(1)}}_{ii} & & \\ & & & S^{(2)} & A^{(2,1)} \\ & & & A^{(1,2)} & S^{(1)} \end{pmatrix} \tag{5}$$

where the matrices L and M are the accumulated Gauss transforms:

$$L = L_{(L)} \cdots L_{(1)} = L^{(n)} \cdots L^{(1)} \qquad ; \qquad M = M_{(L)} \cdots M_{(1)} = M^{(n)} \cdots M^{(1)}$$

Finally, by setting

$$\hat{A}^{(0)} = \begin{pmatrix} S^{(2)} & A^{(2,1)} \\ A^{(1,2)} & S^{(1)} \end{pmatrix} \qquad ; \qquad D = \operatorname{diag}\{{\hat{A}^{(n)}}_{ii}, \ldots, {\hat{A}^{(1)}}_{ii}, \hat{A}^{(0)}\} \tag{6}$$

we obtain the desired LDM^t factorization.

3.3. Construction and Application Cost

Let us start by analyzing the construction cost. In order to make the discussion precise, let us consider a finite element discretization on a structured 2D grid, see Figure 4. When the size of the problem is increased through h-refinements, the dimension of each Schur complement is proportional to $N^{1/2}$. On the other hand, in the case of pure p-enrichments, the dimension is proportional to N. Since pure p-enrichments are not a realistic refinement strategy[1], we limit our analysis to the case of h-refinements.

In [11], we described a procedure that allows to process a non-leaf box on level ℓ within $\mathrm{O}(n_\ell\, k_\ell)$ operations. Here $n_\ell = \mathrm{O}((N/2^\ell)^{1/2})$ is the number of boundary dof's of the box, and k_ℓ is the rank of the off-diagonal block of the associated Schur complement. We hypothesize an asymptotic dependency of the type:

$$k_\ell = k_\ell(n_\ell) = \mathrm{O}({n_\ell}^{\alpha} \log^{\beta}(n_\ell))$$

for some parameters α, β. Thus, the construction cost across the tree levels is

$$\begin{array}{ll} \text{level } L & 2^L\,\mathrm{O}\big(m^3\big) = \mathrm{O}\big(N\big) \\ \dots & \\ \text{level } \ell & 2^\ell\,\mathrm{O}\big((N/2^\ell)^{\alpha+1/2} \log^{2\beta}(N)\big) = 2^{\ell(1/2-\alpha)}\,\mathrm{O}\big(N^{\alpha+1/2} \log^{2\beta}(N)\big) \end{array}$$

which implies a total cost

$$\begin{array}{ll} \alpha = 1/2 & \texttt{cost} = \mathrm{O}\big(N \log^{2\beta+1}(N)\big) \\ \alpha = 1/2+\varepsilon & \texttt{cost} = \mathrm{O}\big(N^{1+\varepsilon} \log^{2\beta}(N)\big) \end{array}$$

In the case $\beta = 0$ and $\alpha < 1/2$, the construction cost reduces to linear, since it is dominated by the cost of processing the leaf-boxes. Numerical studies presented in [11] show that $k_\ell \sim C(p)$ for the Laplace operator, while $k_\ell \sim {n_\ell}^{1/2} \log n_\ell$ for the Helmholtz operator, with no clear dependency on p for low orders. We conclude

[1] Industrial level applications employ either h-refinements on a linear mesh or, at best, h-refinements on a p-mesh.

that

$$\begin{aligned} &\text{2D Laplace:} && \texttt{cost} = \mathrm{O}\big(N\big) \\ &\text{2D Helmholtz:} && \texttt{cost} = \mathrm{O}\big(N \log^3(N)\big) \end{aligned}$$

Finally, the application of $LDM^t\, x = y$ is equivalent to the three solves:

$$L\, y_2 = y \qquad , \qquad D\, y_1 = y_2 \qquad , \qquad M^t\, x = y_1$$

Each Gauss transform is an atomic matrix and, consequently, its inversion is trivial. The action of the factorization can be computed efficiently because of the ability to determine the action of each inverse of $\hat{A}^{(\cdot)}{}_{ii}$ and $\hat{A}^{(0)}$ as described in Step 3 of Algorithm 1. Let us recall that the action of an HSS matrix of size n and off-diagonal rank k can be computed within $\mathrm{O}(nk)$ complexity. Let σ be a box on level ℓ with chid-boxes $\{\mu, \nu\}$. Under assumption that all sub-matrices of $\hat{A}^{(\mu,\nu)}$ and $\hat{A}^{(\nu,\mu)}$ are sparse with band b, the cost of Step 3 is $\mathrm{O}\big((N/2^\ell)^{1/2}(b + k_\ell)\big)$. This is also the application cost of $A^{(\sigma)}{}_{bi}$ or $A^{(\sigma)}{}_{ib}$. As previously, let us assume a dependency of the type

$$k_\ell = k_\ell(n_\ell) = \mathrm{O}(n_\ell{}^\alpha\, \log^\beta(n_\ell))$$

for some parameters α and β. The application at the leaf level is

$$2^L\, \mathrm{O}\big(m^2\big) = \mathrm{O}\big(N\big)$$

while the for level ℓ is

$$2^\ell\, \mathrm{O}\Big(\big(N/2^\ell\big)^{1/2}\big(b + (N/2^\ell)^{\alpha/2} \log^\beta(N)\big)\Big) = 2^{\ell/2}\, \mathrm{O}\big(N^{1/2} b\big) + 2^{\ell\frac{1-\alpha}{2}}\mathrm{O}\big(N^{\frac{\alpha+1}{2}} \log^\beta(N)\big)$$

If we sum over all levels, the total application cost is:

$$\begin{aligned} &\alpha = 1 && \texttt{cost} = \mathrm{O}\big(N\big) + \mathrm{O}\big(Nb\big) + \mathrm{O}\big(N \log^{\beta+1}(N)\big) \\ &\alpha < 1 && \texttt{cost} = \mathrm{O}\big(N\big) + \mathrm{O}\big(Nb\big) + \mathrm{O}\big(N \log^{\beta}(N)\big) \end{aligned}$$

3.4. Preconditioner Update

Let us suppose that we are given a sequence of discrete operators $\{A_i\}_{i \geq 0}$ which, in vague terms, are “slowly varying.” While a rigorous argument would involve matrix perturbation theory, we keep this discussion informal. A typical example of this situation arises for subsequent discretizations of a boundary value problem, subject to small modifications of its coefficients. When this is the case, instead of computing the factorization of A_i directly, we would like to simply update the factorization of A_{i-K} for some K, hopefully large.

An updating strategy can be pursed as follows. Let A_0 be the seed matrix for which we have computed the factorization LDM^t. Then a factorization of $A := A_0 + \Delta$ is immediately obtained as:

$$A = A_0 + \Delta = LDM^t + \Delta = L\big(D + L^{-1}\Delta M^{-t}\big)M^t$$

and, proceeding formally, we obtain

$$A^{-1} = M^{-t}\big(D + L^{-1}\Delta M^{-t}\big)^{-1}L^{-1} = M^{-t}\big(I + D^{-1}L^{-1}\Delta M^{-t}\big)^{-1}D^{-1}L^{-1}$$

Under the assumption $\varrho(D^{-1}L^{-1}\Delta M^{-t}) < 1$, where $\varrho(\cdot)$ indicates the spectral radius, the term in parenthesis is indeed invertible, and we can carry out such inversion through a Neumann series:

$$\begin{aligned} A^{-1} &= M^{-t}\big(I - D^{-1}L^{-1}\Delta M^{-t} + (D^{-1}L^{-1}\Delta M^{-t})^2 - \cdots\big)D^{-1}L^{-1} \\ &= P_0 - P_0\Delta P_0 + P_0\Delta P_0\Delta P_0 - \cdots \end{aligned}$$

Here we set $P_0 = M^{-t}D^{-1}L^{-1}$. For $d \geq 1$, we define:

$$P_d = \sum_{i=0}^{d}(-)^i P_0(\Delta P_0)^i$$

The action of P_d can be evaluated as described in Algorithm 2. It is important to notice that P_d is fully determined once P_0 has been constructed. Furthermore, since Δ is the difference of two sparse matrices, the cost of applying P_d is approximately $d+1$ times the cost of applying P_0. We indicate the construction of P_0 with the notation $P_0 = \texttt{prec}(A, \varepsilon, k, m, m_{\text{HSS}})$. Here $k = k(n)$ is the anticipated rank of an off-diagonal block of dimension n of any Schur complement, ε is its compression threshold in the Frobenius norm, and m_{HSS} is the maximum dimension of an uncompressed diagonal block, see [11] for more details.

4. Numerical Results

As anticipated in Section 1, Topology Optimization is an iterative method, used mainly for PDE's constrained optimization problems, to create highly optimized de-

```
input  : $P_0$, $\Delta$, $z$, $d$
output: $y$

Store $x = z$;
for $i = 1, \dots, d$ do
    update $z = x - \Delta P_0\, z$;
end
compute $y = P_0\, z$;
```

Algorithm 2: Computation of the action $y = P_d\, z$.

signs for specific purposes. The objective is to determine a distribution of material that fulfills a specific task in a locally optimal manner, without the need to enforce any *a priori* restriction on the design topology, see [4]. Although the use of gradient-based techniques results in a significant reduction in the number of iterations needed [23], for many problems the method still requires around 100 to 1000 iterations with a slowly varying design field to recover a locally optimal and physically admissible design. Consequently, the governing equations for the problem under consideration must be solved a large number of times for a slowly changing material distribution. When PDE problems of several millions of degrees of freedom are considered, as it is often the case for actual industrial-scale applications, their solutions through traditional direct techniques become infeasible. This raises the interest in using highly scalable parallel iterative techniques. For physical problems with rapidly varying high contrast material parameters governed by the Helmholtz equation, like acoustic, electromagnetic and structural vibration problems, apart from the preconditioner proposed in [11], we are not aware of any effective iterative technique. Therefore it is of interest to investigate the performance of this preconditioner in the context of Topology Optimization.

We consider the optimization problem of a 2D acoustic cavity Ω containing an array of square sub-domains, whose union Ω_{d} is denoted as the design domain. The formulation is based on [8] and [7]. The resulting optimization problem is highly non-convex. The objective is to minimize the average sound pressure in a small target sub-domain $\Omega_{\mathrm{op}} \subset \Omega$, by introducing, or removing, material acting as a hard wall in Ω_{d}. The boundary of the cavity $\partial\Omega$ is taken to be perfectly reflecting, except for a small section δP where a pure tone with vibrational velocity U is excited through a non-zero Neumann condition, see Figure 5. The pressure field p is governed by the following Helmholtz problem:

$$\operatorname{div}\left(\frac{1}{\rho}\,\nabla p\right) + \frac{\omega^2}{\kappa} p = 0 \qquad \text{in } \Omega \tag{7a}$$

$$\frac{1}{\rho}\,\nabla p \cdot \boldsymbol{n} = 0 \qquad \text{on } \operatorname{int}(\partial\Omega \setminus \delta P) \tag{7b}$$

$$\frac{1}{\rho}\,\nabla p \cdot \boldsymbol{n} = -\mathrm{i}\omega U \qquad \text{on } \delta P \tag{7c}$$

Here i denotes the imaginary unit, and ω is the angular frequency. As for the pressure, both the density ρ and bulk modulus κ depend on the spatial position. In fact, the physics of the problem dictates that a given spatial position contains either solid or void. When the contrast in material parameters is chosen such that no transverse waves of significant amplitude are excited in the solid regions, the boundary of the material acts nearly identically to a perfectly reflecting boundary, i.e., a hard wall reflecting the sound wave with insignificant transmission. The parameters for the void regions are taken to be those of air at standard conditions[2], i.e., $\rho_{\mathrm{air}} = 1.204\,\mathrm{kg\,m^{-3}}$, $\kappa_{\mathrm{air}} = 141.921 \cdot 10^3\,\mathrm{Pa}$, while the parameters for the solid regions are taken to be those of aluminum, i.e., $\rho_{\mathrm{alu}} = 2643\,\mathrm{kg\,m^{-3}}$, $\kappa_{\mathrm{alu}} = 6.87 \cdot 10^{10}\,\mathrm{Pa}$. The excitation frequency $f = 2750$ Hz corresponding to the wave number $k = 2\pi f/c, c = \sqrt{\kappa_{\mathrm{air}}/\rho_{\mathrm{air}}}$ and $U = 0.01\,\mathrm{m/s}$ are used.

In order to employ a continuous optimization approach, an auxiliary field ξ, such that $0 \le \xi(\boldsymbol{x}) \le 1$ when $\boldsymbol{x} \in \Omega_{\mathrm{d}}$, and $\xi(\boldsymbol{x}) = 0$ when $\boldsymbol{x} \in \Omega \setminus \Omega_{\mathrm{d}}$, is introduced to interpolate between the inverse material parameters of solid and air. Thus, a location where $\xi = 1$ consists of solid material, while a location where $\xi = 0$ is occupied by air. Since a straightforward application of this strategy often results in fragmented and physically inadmissible designs, we apply a smoothing and a projection operator to the auxiliary field, along with a continuation scheme on the projection strength, see [16, 25, 7]. For ease of notation, let ξ indicate the auxiliary field after smoothing and projection. The minimization of the average of p over Ω_{op} may be stated as the continuous optimization problem,

$$\bar{\xi} = \underset{\xi}{\operatorname{argmin}}\left\{ \Phi(\xi) := \int_{\Omega_{\mathrm{op}}} \left| p\big(\boldsymbol{x}, \xi(\boldsymbol{x})\big) \right|^2 \right\}$$

where p is obtained by solving (7) for a given realization of ξ. Adjoint sensitivity

[2] 0 % humidity, 20 °C temperature, and 1 atm background pressure

analysis is applied to obtain the gradient of the objective function with respect to ξ, see e.g. [24].

The PDE model problem is discretized using a structured quadrilateral mesh consisting of 150×150 linear finite elements, which yields 22801 dof's. Similarly, the auxiliary field ξ is approximated by a piecewise-constant function, so that each finite element is associated to a single variable, termed "design variable." The process involves 100 optimization iterations, and the evolution of the design variables and pressure field is illustrated for three iterations in Figure 6. The preconditioner is constructed or updated using the strategy described in Algorithm 3. We remark that such algorithm terminates, provided that $n \leq n_{\max}$ when $A = A_{\text{seed}}$. We set $k \sim n^{1/2} \log n$, $m = 100$, $m_{\text{HSS}} = 10$, $\varepsilon = 10^{-6}$ for the preconditioner construction, and use $n_{\max} = 40$, and $d = 2$ for the updating strategy. For comparison, we also ran the same experiment with $d = 0, 1$. Since the compression to HSS format relies on Gauss matrices, simulations for different seeds of the random number generator were performed. The results are illustrated in Figure 7. The behavior is qualitatively similar for different instantiations of the random number generator. In terms of GMRES iterations, the updating strategy is working convincingly. In fact, as compared to $d = 0$, namely no update, the choice $d = 2$ results in a substantial decrease of iterations needed by the solver. Nevertheless, given the increased cost in the application of the preconditioner for higher values of d, we shall remark that the choice $d = 2$ does not necessarily lead to a faster execution time. A rigorous investigation would require to compare the construction and application time. Since at the moment we do not have an aggressively optimized implementation available, we postpone this discussion to future work. Finally, we observe that, in order to process 100 optimization steps, only 6 or 7 recomputes of the preconditioner were needed.

5. Conclusions and Outlook

In this work, we review the construction of the preconditioner proposed in [11], sharpen our theoretical understanding of the methodology, and provide accurate cost estimates for different scenarios. The main novel contribution is a strategy to apply the preconditioner to time-varying discretizations. Specifically, we discuss an application to Topology Optimization of an acoustic cavity, which requires multiple solutions of a notoriously challenging Helmholtz problem with strong material discontinuities. Although we confined our numerical results to one case of modest size, they show the robustness of the preconditioner and the updating strategy. As for future directions of research, we would like to investigate the interplay between partitioning schemes and distributions of material. As the size of the problem grows, we expect this issue to become increasingly important. At the same time, we believe that it could be addressed in a black-box fashion through the use an algebraic partitioner and weighted graphs.

References

[1] A. Aminfar and E. Darve. A fast, memory efficient and robust sparse preconditioner based on a multifrontal approach with applications to finite-element matrices. *International Journal for Numerical Methods in Engineering*, 2016.

[2] M. Bebendorf. *Hierarchical matrices*. Springer, 2008.

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

[4] M.P. Bendsøe and O. Sigmund. *Topology Optimization*. Springer, 2003.

[5] 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.

[6] S. Chandrasekaran, M. Gu, and T. Pals. A fast *ulv* decomposition solver for hierarchically semiseparable representations. *SIAM Journal on Matrix Analysis and Applications*, 28(3):603–622, 2006.

[7] R.E. Christiansen, B.S. Lazarov, J.S. Jensen, and O. Sigmund. Creating geometrically robust designs for highly sensitive problems using topology optimization. *Structural and Multidisciplinary Optimization*, 52(4):737–754, 2015.

[8] M.B. Dühring, J.S. Jensen, and O. Sigmund. Acoustic design by topology optimization. *Journal of Sound and Vibration*, 317:557–575, 2008.

[9] B. Engquist and H. Zhao. Approximate separability of green's functions for high frequency helmholtz equations. Technical report, CALIFORNIA UNIV. LOS ANGELES DEPT. OF MATHEMATICS, 2014.

[10] Björn Engquist and Lexing Ying. Fast directional multilevel algorithms for oscillatory kernels. *SIAM Journal on Scientific Computing*, 29(4):1710–1737, 2007.

[11] P. Gatto and J.S. Hesthaven. Efficient preconditioning of *hp*-fem matrices by hierarchical low-rank compression. 2016.

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

[13] A. Gillman and P.G. Martinsson. An $\mathrm{O}(n)$ 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.

[14] A. Gillman, P.M. Young, and P.G. Martinsson. A direct solver with $\mathrm{O}(n)$ complexity for integral equations on one-dimensional domains. *Frontiers of Mathematics in China*, 7(2):217–247, 2012.

[15] L. Grasedyck, R. Kriemann, and S. LeBorne. Domain decomposition based $\mathcal{H}$-lu preconditioning. *Numerische Mathematik*, 112(4):565–600, 2009.

[16] J.K. Guest, J.H. Prvost, and T. Belytschko. Achieving minimum length scale in topology optimization using nodal design variables and projection functions. *International Journal for Numerical Methods in Engineering*, 61:238–254, 2004.

[17] W. Hackbusch. A sparse matrix arithmetic based on $\mathcal{H}$-matrices. part i: Introduction to $\mathcal{H}$-matrices. *Computing*, 62(2):89–108, 1999.

[18] W. Hackbusch, B. Khoromskij, and S. Sauter. On $\mathcal{H}^2$-matrices. *Lectures on Applied Mathematics*, pages 9–29, 2000.

[19] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on scientific Computing*, 20(1):359–392, 1998.

[20] 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.

[21] P.G. Martinsson and V. Rokhlin. A fast direct solver for scattering problems involving elongated structures. *Journal of Computational Physics*, 221(1):288–302, 2007.

[22] Z. Sheng, P. Dewilde, and S. Chandrasekaran. Algorithms to solve hierarchically semi-separable systems. In *System theory, the Schur algorithm and multidimensional analysis*, pages 255–294. Springer, 2007.

[23] Ole Sigmund. On the usefulness of non-gradient approaches in topology optimization. *Structural and Multidiciplinary Optimization*, 43:589596, 2011.

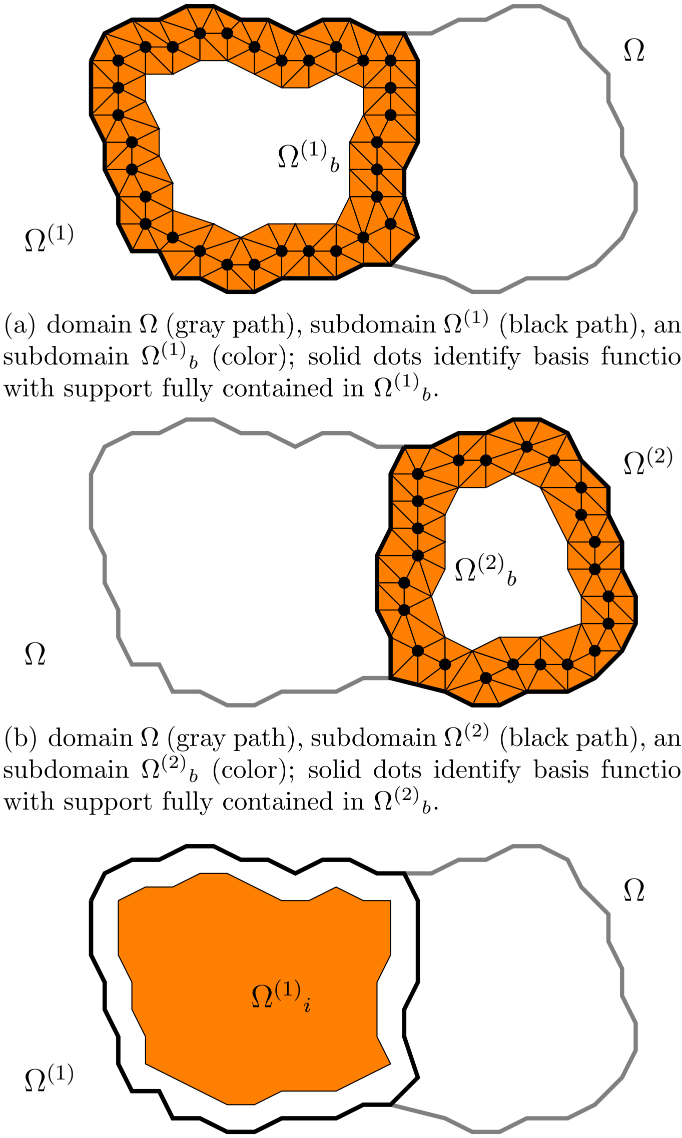
[24] Daniel A. Tortorelli and Panagiotis Michaleris. Design sensitivity analysis: Overview and review. *Inverse Problems in Engineering*, 1:71105, 1994.

[25] F. Wang, B.S. Lazarov, and O. Sigmund. On projection methods, convergence and robust formulations in topology optimization. *Structural Multidiciplinary Optimization*, 43:767–784, 2011.

[26] J. Xia. Efficient structured multifrontal factorization for general large sparse matrices. *SIAM Journal on Scientific Computing*, 35(2):A832–A860, 2013.

[27] 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.

[28] J. Xia, S. Chandrasekaran, M. Gu, and X.S. Li. Fast algorithms for hierarchically semiseparable matrices. *Numerical Linear Algebra with Applications*, 17(6):953–976, 2010.

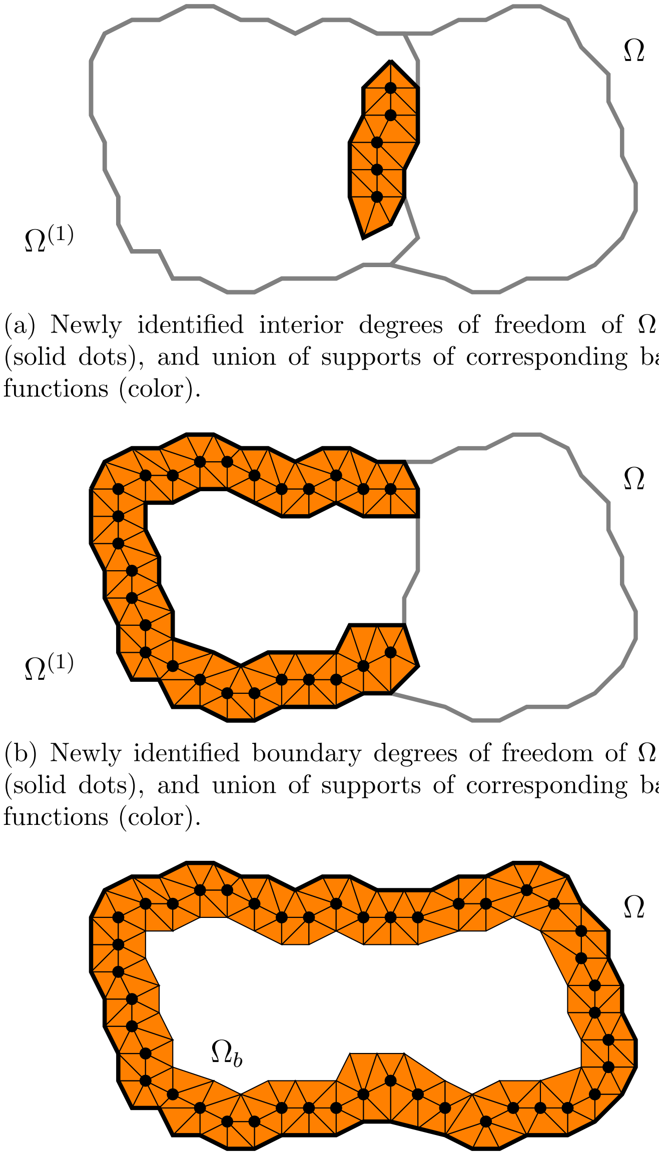


(a) domain Ω (gray path), subdomain $\Omega^{(1)}$ (black path), and subdomain $\Omega^{(1)}{}_b$ (color); solid dots identify basis function with support fully contained in $\Omega^{(1)}{}_b$.

(b) domain Ω (gray path), subdomain $\Omega^{(2)}$ (black path), and subdomain $\Omega^{(2)}{}_b$ (color); solid dots identify basis function with support fully contained in $\Omega^{(2)}{}_b$.

(c) Domain Ω (gray path), subdomain $\Omega^{(1)}$ (black path), and subdomain $\Omega^{(1)}{}_i$ (color).

Figure 1: Subdomains $\Omega^{(\cdot)}$, $\Omega^{(\cdot)}{}_b$, and $\Omega^{(\cdot)}{}_i$ in the case of a first order, conforming finite element approximation. Each degree of freedom, or basis function, is uniquely identified with a mesh node.



(a) Newly identified interior degrees of freedom of $\Omega^{(1)}{}_b$ (solid dots), and union of supports of corresponding basis functions (color).

(b) Newly identified boundary degrees of freedom of $\Omega^{(1)}{}_b$ (solid dots), and union of supports of corresponding basis functions (color).

(c) Newly identified boundary degrees of freedom of $\Omega^{(1)}{}_b$ and $\Omega^{(2)}{}_b$ (solid dots), and union of supports of corresponding basis functions Ω_b (color).

Figure 2: Interior and boundary degrees of freedom of subdomain $\Omega^{(1)}{}_b$, and merging of boundary degrees of freedom of subdomain $\Omega^{(1)}{}_b$ and $\Omega^{(2)}{}_b$. A first order, conforming finite element approximation is considered. Each degree of freedom, or basis function, is uniquely identified with a mesh node.

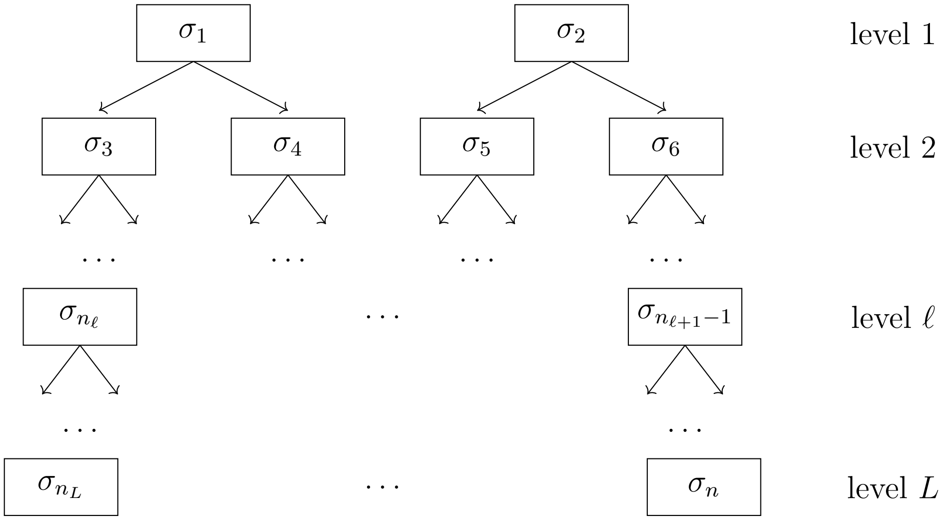
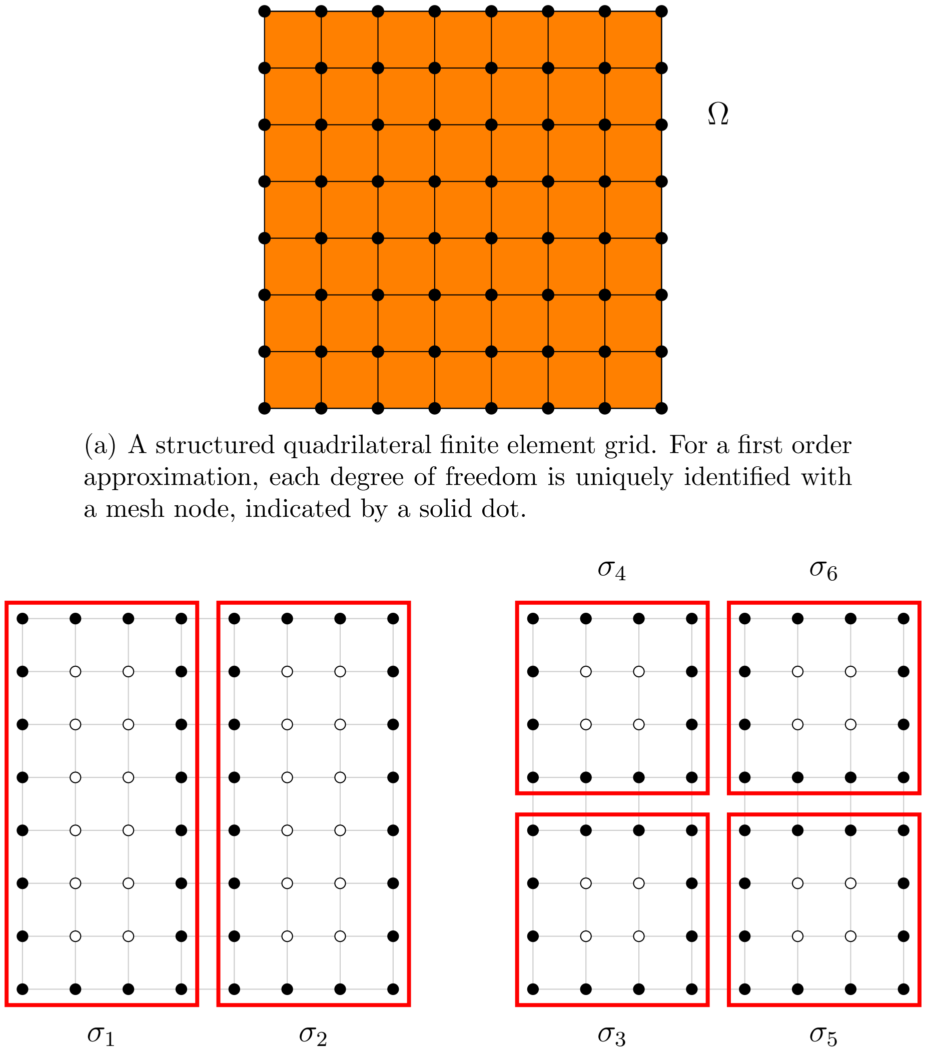


Figure 3: Enumeration of binary tree of boxes for dof's partition



(a) A structured quadrilateral finite element grid. For a first order approximation, each degree of freedom is uniquely identified with a mesh node, indicated by a solid dot.

(b) Repartitioning into 2 subdomains. Each boundary contains $\sim (N/2)^{1/2}$ degrees of freedom, indicated by solid dots.

(c) Repartitioning into 4 subdomains. Each boundary contains $\sim (N/4)^{1/2}$ degrees of freedom, indicated by solid dots.

Figure 4: Partitioning of a first order finite element approximation on a structured quadrilateral mesh with N degrees of freedom.

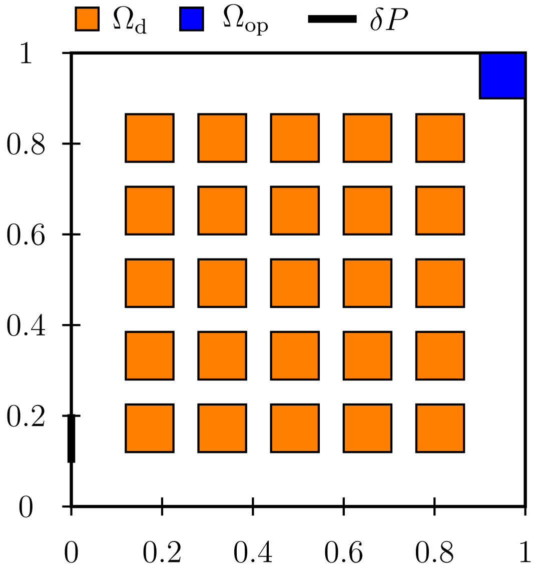
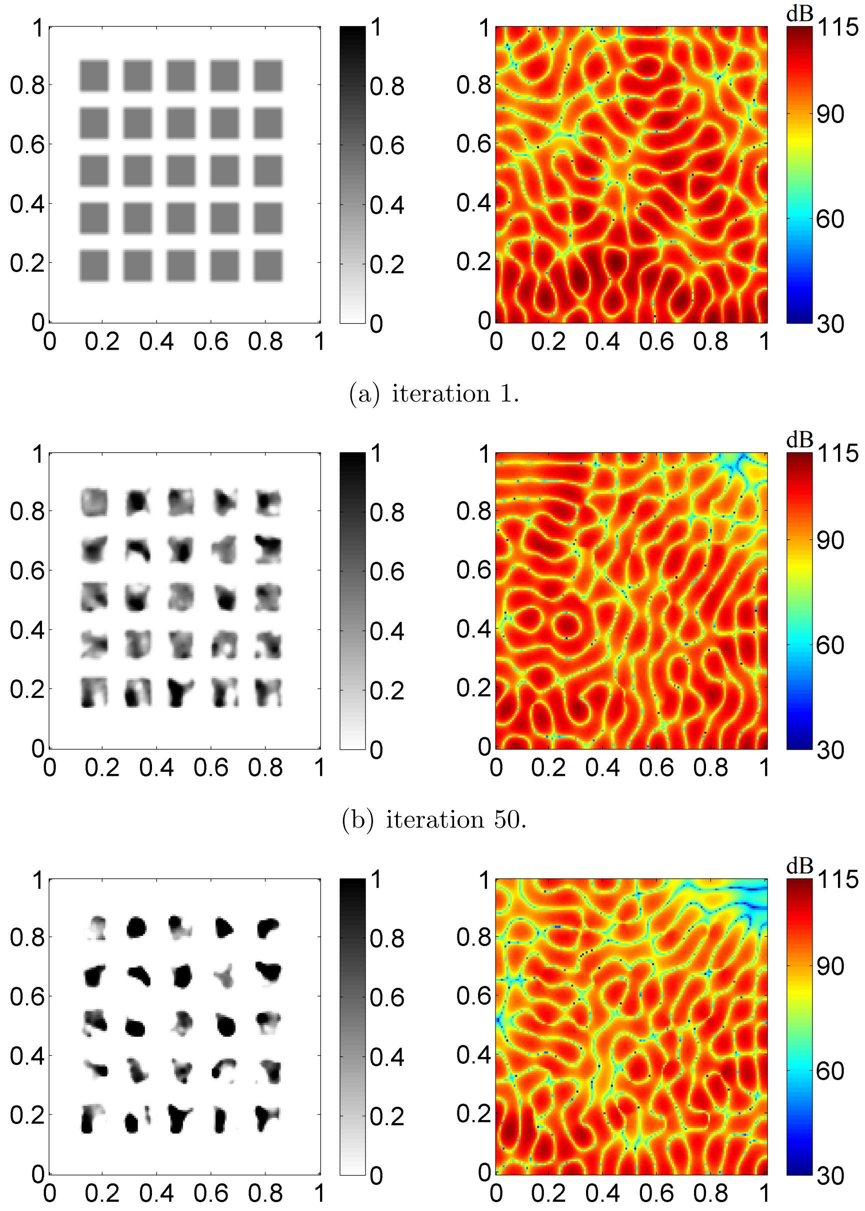


Figure 5: TopOpt problem setup. Acoustic cavity $\Omega = [0,1]^2$, optimization domain Ω_op, design domain, Ω_d. Problem is driven by a forced vibration imposed on δP. The remaining portion of the boundary is treated as a reflecting boundary.

```
input  : $\varepsilon$, $k$, $m$, $m_{\mathrm{HSS}}$, $\{A_0, \ldots, A_K\}$, $\{b_0, \ldots, b_K\}$, $n_{\max}$, $d$
output: $\{x_0, \ldots, x_K\}$
step $= 0$;
load $= 0$;
comp $= 0$;
$x_{\mathrm{guess}} = b_0$;
/* Do until last optimization step */
while step $\leq K$ do
    /* Load new matrix */
    if load $= 0$ then
        load $A = A_{\mathtt{step}}$, $b = b_{\mathtt{step}}$;
        set load $= 1$;
    end
    /* Compute preconditioner */
    if comp $= 0$ then
        construct $P_0 = \mathtt{prec}(A, \varepsilon, k, m, m_{\mathrm{HSS}})$;
        set $A_{\mathrm{seed}} = A$;
        set comp $= 1$;
    end
    /* GMRES solve */
    compute $[x, n] = \mathtt{gmres}(A, b, P_d, x_{\mathrm{guess}})$;
    /* Unsuccessful solve */
    if $n > n_{max}$ then
        set comp $= 0$;
    else
        store $x_{\mathtt{step}} = x$;
        set $x_{\mathrm{guess}} = x$;
        set load $= 0$;
        increment step $=$ step $+ 1$;
    end
end
```

Algorithm 3: Preconditioner recompute/update strategy for topology optimization.

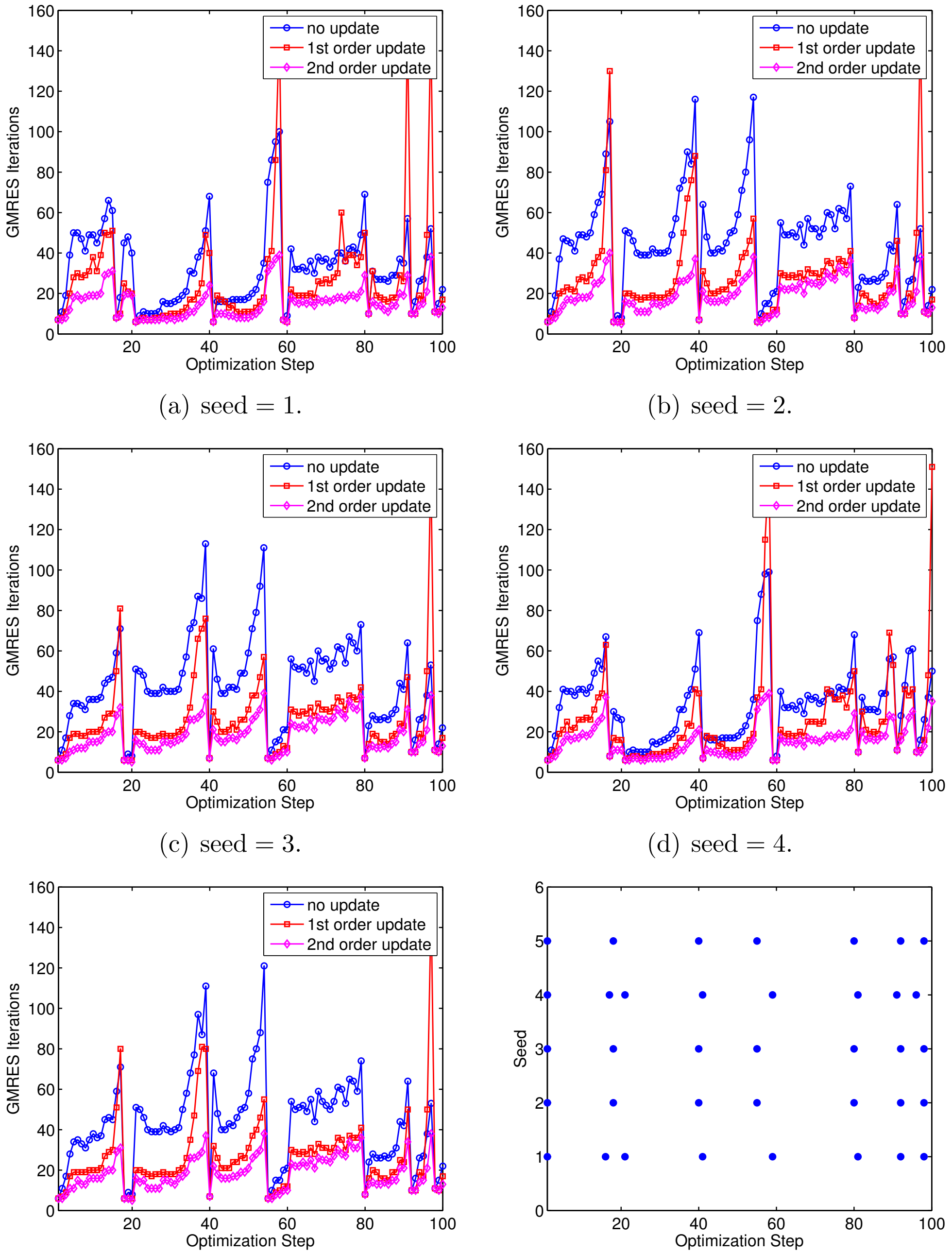


(a) iteration 1.

(b) iteration 50.

(c) iteration 100.

Figure 6: Evolution of design variables (left) and pressure distribution [dB SPL ref 20μPa] (right) at selected iterations for a topology optimization problem with 100 iterations. The physics is discretized using 150 × 150 quadrilateral first order finite elements.



(a) seed = 1.

(b) seed = 2.

(c) seed = 3.

(d) seed = 4.

(e) seed = 5.

(f) recomputes indicated by a solid dot.

Figure 7: Behavior of preconditioned-GMRES ($k \sim n^{1/2} \log n$, $m = 100$, $m_{\text{HSS}} = 10$, $\varepsilon = 10^{-6}$) for 100 steps of a topology optimization problem, for 5 seeds of the random number generator.