

Solving an Elliptic PDE Eigenvalue Problem via Automated Multi-Level Substructuring and Hierarchical Matrices

Peter Gerds* and Lars Grasedyck*

Bericht Nr. 390

März 2014

Key words: automated multi-level substructuring,
hierarchical matrices,
elliptic PDE eigenvalue problem

AMS Subject Classifications: 65F15, 65F30, 65F50,
65H17, 65N25, 65N55

**Institut für Geometrie und Praktische Mathematik
RWTH Aachen**

Templergraben 55, D-52056 Aachen (Germany)

* Institut für Geometrie und Praktische Mathematik, RWTH Aachen, Templergraben 55, 52056 Aachen, Germany. Email: {gerds,lgr}@igpm.rwth-aachen.de

Solving an Elliptic PDE Eigenvalue Problem via Automated Multi-Level Substructuring and Hierarchical Matrices

Peter Gerds*, Lars Grasedyck*

March 7, 2014

We propose a new method for the solution of discretised elliptic PDE eigenvalue problems. The new method combines ideas of domain decomposition, as in the *automated multi-level substructuring* (short AMLS), with the concept of *hierarchical matrices* (short \mathcal{H} -matrices) in order to obtain a solver that scales almost linearly (linear up to logarithmic factors) in the size of the discrete space. Whereas the AMLS method is very effective for PDEs posed in two dimensions, it is getting very expensive in the three-dimensional case, due to the fact that the interface coupling in the domain decomposition requires dense matrix operations. We resolve this problem by use of data-sparse hierarchical matrices. In addition to the discretisation error our new approach involves a projection error due to AMLS and an arithmetic error due to \mathcal{H} -matrix approximation. A suitable choice of parameters to balance these errors is investigated in examples.

Mathematics Subject Classification (2000) 65F15, 65F30, 65F50, 65H17, 65N25, 65N55

Keywords Automated multi-level substructuring, hierarchical matrices, elliptic PDE eigenvalue problem

1 Introduction

A very efficient approach to solve an elliptic PDE eigenvalue problem is the so-called *automated multi-level substructuring* (short AMLS) method. AMLS is a substructuring method which was mainly developed by Bennighof and co-authors [6, 22, 4] and is based on the classical *component mode synthesis* (short CMS).

*Institut für Geometrie und Praktische Mathematik, RWTH Aachen, Templergraben 55, 52056 Aachen, Germany. Email: {gerds,lgr}@igpm.rwth-aachen.de.

The CMS is as well a substructuring method which was already developed in the 1960s to solve large scale eigenvalue problems in structural engineering analysis. The method was firstly described by Hurty [21] and further improved by Craig and Bampton [10]. During the years CMS became very popular and was studied by many researchers, e.g. in [8, 9] a mathematical analysis of CMS is given and in [26] an overview over different versions of CMS.

The single-level substructuring performed in CMS is extended in AMLS to the multi-level case. The idea in AMLS is to partition the spatial domain of the PDE eigenvalue problem recursively into several subdomains. On each of these subdomains similar eigenvalue problems are defined which are typically small and easy to solve. From each of these subproblems suitable solutions are selected which are meant to represent the global problem on the subdomain. All selected solutions together form a subspace. The global eigenvalue problem is projected onto this subspace and a *reduced eigenvalue problem* is obtained which is typically much smaller than the original problem and correspondingly much easier to solve. Finally, the eigenpairs of the reduced eigenvalue problem deliver the sought eigenpair approximations of the global eigenvalue problem.

In [5, 22, 23] AMLS has proven to be very effective for solving large-scale eigenvalue problems arising in structural engineering analysis. Especially when a large number of eigenpair approximations is required AMLS is more effective than classical approaches using algebraic eigensolvers which are coupled with a preconditioner or a linear solver (cf. [20]). The big advantage of AMLS is that it computes several eigenpairs at once whereas the computational costs of classical approaches are at least linear in the number of sought eigenpairs. A very popular of such a classical approach is the *shift-invert block Lanczos* (short SIL) algorithm [16] which is commonly used in structural engineering. Kropp and Heiserer presented breakthrough calculations in [23]. They benchmarked AMLS against SIL within a vibro-acoustic analysis of an automobile body and could show that AMLS running on a commodity workstation is several times faster than SIL running on a supercomputer.

When AMLS is applied to a discrete eigenvalue problem it computes only eigenpair approximations whereas SIL computes numerically almost exact eigenpairs. This seems to be disadvantageous, however, in our setting a discrete eigenvalue problem results always from a finite element discretisation of a continuous eigenvalue problem. Correspondingly all computed eigenpairs of the discrete problem are related to a discretisation error. As long as the projection error caused by AMLS is of the same order as the discretisation error the computed eigenpair approximations of AMLS are of comparable quality as the eigenpairs computed by SIL or some other classical approach.

Although AMLS has proven to be very effective, one problem is the computation of the interface eigenvalue problem via dense matrix operations. In the three-dimensional case the complexity is dominated by this part.

In this paper we present a new approach called \mathcal{H} -AMLS which is a combination of the AMLS method and \mathcal{H} -matrices. \mathcal{H} -matrices [18, 19] are a data-sparse approximation of dense matrices which e.g. result from the inversion [3] or the *LU*-factorisation [14] of the stiffness matrix from the finite element discretisation of an elliptic partial differential operator. The big advantage of \mathcal{H} -matrices is that they allow matrix algebra in almost linear complexity [13]. In the new method this fast \mathcal{H} -matrix algebra is used to compute the reduced eigenvalue

problem and thus \mathcal{H} -AMLS is well-suited for three-dimensional problems.

The remainder of the paper is organised as follows: In Section 2 the elliptic eigenvalue problem and the underlying problem setting is introduced. After this, in Section 3, we give a description of the classical AMLS method. We explain and motivate the method first in a continuous setting and then describe it in an algebraic setting to show how AMLS is applied in practice. In Section 4, we outline why the classical AMLS method is getting expensive in the three-dimensional case and give in Section 5 a short introduction to \mathcal{H} -matrices. Finally, in Section 6, we present the new \mathcal{H} -AMLS method and provide in Section 7 numerical results of \mathcal{H} -AMLS applied to a three-dimensional problem.

2 Problem Description

In this paper we want to solve the continuous eigenvalue problem

$$\begin{cases} Lu = \lambda u & \text{in } \Omega, \\ u = 0 & \text{on } \partial\Omega \end{cases} \quad (1)$$

where L is a uniformly elliptic second order partial differential operator in divergence form

$$Lu = -\operatorname{div}(A\nabla u) = -\sum_{i,j=1}^d \frac{\partial}{\partial x_i} \left(a_{ij} \frac{\partial}{\partial x_j} u \right)$$

where the coefficients of the symmetric matrix $A := (a_{ij})_{i,j=1}^d$ are $L^\infty(\Omega)$ -functions and Ω is a d -dimensional domain ($d = 2, 3$) with a Lipschitz boundary $\partial\Omega$. In weak formulation (1) can be expressed as

$$\begin{cases} \text{find } (\lambda, u) \in \mathbb{R} \times H_0^1(\Omega) \text{ such that} \\ a(u, v) = \lambda (u, v)_0 \quad \forall v \in H_0^1(\Omega) \end{cases} \quad (2)$$

where $a(u, v) := \int_{\Omega} \nabla u^T A \nabla v \, dx$ is a symmetric, coercive bilinear form and $(u, v)_0 := \int_{\Omega} u \bar{v} \, dx$ is the inner product of $L^2(\Omega)$.

The continuous eigenvalue problem (2) possesses a countable family of eigensolutions

$$(\lambda_j, u_j)_{j=1}^{\infty} \in \mathbb{R}_{>0} \times H_0^1(\Omega) \quad (3)$$

where the eigenvalues λ_j are positive and where we can assume that the eigensolutions are arranged in such a way that $\lambda_j \leq \lambda_{j+1}$ holds.

We approximate solutions of the continuous eigenvalue problem by discretisation. Using an N -dimensional finite element space denoted by $V_N \subset H_0^1(\Omega)$ and spanned by its basis functions $(\varphi_i^{(N)})_{i=1}^N$ the continuous eigenvalue problem (2) is discretised by

$$\begin{cases} \text{find } (\lambda^{(N)}, x^{(N)}) \in \mathbb{R} \times \mathbb{R}^N \text{ with} \\ K x^{(N)} = \lambda^{(N)} M x^{(N)} \end{cases} \quad (4)$$

where the stiffness matrix

$$K := \left(a(\varphi_i^{(N)}, \varphi_j^{(N)}) \right)_{i,j=1}^N \in \mathbb{R}^{N \times N} \quad (5)$$

and the mass matrix

$$M := \left((\varphi_i^{(N)}, \varphi_j^{(N)})_0 \right)_{i,j=1}^N \in \mathbb{R}^{N \times N} \quad (6)$$

are both sparse, symmetric and positive definite. The eigenvalues of (4) are positive and the corresponding eigenpairs $(\lambda_j^{(N)}, x_j^{(N)})_{j=1}^N \in \mathbb{R}_{>0} \times \mathbb{R}^N$ can be arranged in such a way that $\lambda_j^{(N)} \leq \lambda_{j+1}^{(N)}$ holds.

From eigenvalue approximation theory it follows that the discrete eigenpairs of (4) are approximating the continuous eigensolutions of (2). More precisely, cf. [17], it holds $\lambda_j^{(N)} \rightarrow \lambda_j$ for $N \rightarrow \infty$ and a subsequence of $(u_j^{(N)})_{N=1}^\infty$ is converging in $H_0^1(\Omega)$ against u_j , where $u_j^{(N)}$ is defined by $u_j^{(N)} := \sum_{i=1}^N (x_j^{(N)})_i \cdot \varphi_i^{(N)} \in V_N$, and we assume that $\|u_j^{(N)}\|_0 = 1$ and $\|u_j\|_0 = 1$ with $\|\cdot\|_0 := (\cdot, \cdot)_0^{1/2}$.

Here it has to be noted that only the smaller eigenvalues λ_j and their corresponding eigenfunctions u_j can be approximated by the finite element space V_N (cf. [2, 25]), because the approximation error increases with increasing eigenvalue.

Correspondingly we are only interested in computing a portion of the eigenpairs of (4), e.g., the first

$$n_{ev} = CN^{1/3} \in \mathbb{N} \quad \text{or} \quad n_{ev} = CN^{1/2} \in \mathbb{N}$$

eigenpairs, for some constant $C > 0$.

Because we are interested in a large number of eigensolutions, the AMLS method is used to solve the eigenvalue problem (2), respectively (4). If the number of sought eigensolutions n_{ev} is rather small, e.g. $n_{ev} = 5$, other approaches like the subspace iteration are better suited.

3 The AMLS method

Although AMLS can be described in a purely algebraic way without any geometry information of the underlying partial differential equation we explain the method first in a continuous setting. In the continuous setting it is easier to understand the idea behind AMLS and why the method is working. After this we will describe AMLS in an algebraic setting to show how the method is used in practice. For ease of understanding we start with the description of a single-level version of AMLS which is extended to a multi-level version afterwards.

3.1 Single-Level Case: Continuous Setting

In the single-level case AMLS is actually a generalisation of the classic CMS. The initial point of AMLS in the continuous setting is the eigenvalue problem (2) which will be denoted as *global eigenvalue problem* in this particular section. In the first step of AMLS the domain Ω is partitioned into two non-overlapping subdomains Ω_1 and Ω_2 which share the interface

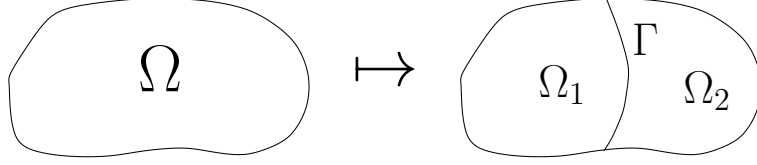


Figure 1: Partitioning of the domain Ω into two non-overlapping subdomains.

$\Gamma := \overline{\Omega}_1 \cap \overline{\Omega}_2$. In Figure 1 an example of such a partitioning is given for a two-dimensional domain.

After this suitable subspaces of $H_0^1(\Omega)$ are defined which are associated with the subdomains Ω_i ($i = 1, 2$) and the interface Γ . For the subdomains Ω_i we define

$$V_{\Omega_i} := \left\{ v \in H_0^1(\Omega) \mid v|_{\Omega \setminus \Omega_i} = 0 \right\}$$

built of all admissible functions which are equal to zero on $\Omega \setminus \Omega_i$, and for Γ we define

$$V_{\Gamma} := \left\{ E_{\Omega} \tau \mid \tau \in H_0^{1/2}(\Gamma) \right\}.$$

Here $H_0^{1/2}(\Gamma)$ denotes the trace space of $H_0^1(\Omega)$ on the interface Γ and $E_{\Omega} \tau \in H_0^1(\Omega)$ is the extension of the trace function $\tau \in H_0^{1/2}(\Gamma)$ which is defined as the solution of the Dirichlet problem

$$\begin{cases} \text{find } E_{\Omega} \tau \in H_0^1(\Omega) \text{ such that} \\ a(E_{\Omega} \tau, v) = 0 \quad \forall v \in \{u \in H_0^1(\Omega) \mid u|_{\Gamma} = 0\}, \\ E_{\Omega} \tau = \tau \quad \text{on } \Gamma. \end{cases}$$

For the three subspaces the following theorem holds:

Theorem 1. *The direct sum*

$$V_{\Omega_1} \oplus V_{\Omega_2} \oplus V_{\Gamma}$$

is an a -orthogonal decomposition of $H_0^1(\Omega)$.

Proof. A proof can be found in [6] in the context of an eigenvalue problem from linear elastodynamics. However, this proof can be applied for generic H^1 -elliptic bilinear forms. \square

In the second step of AMLS we define for each subspace separate eigenvalue problems; for V_{Ω_i} ($i = 1, 2$) the so-called *fixed-interface eigenvalue problem*

$$\begin{cases} \text{find } (\lambda, u) \in \mathbb{R} \times V_{\Omega_i} \text{ such that} \\ a(u, v) = \lambda(u, v)_0 \quad \forall v \in V_{\Omega_i}, \end{cases} \quad (7)$$

and for V_{Γ} the so-called *coupling mode eigenvalue problem*

$$\begin{cases} \text{find } (\lambda, u) \in \mathbb{R} \times V_{\Gamma} \text{ such that} \\ a(u, v) = \lambda(u, v)_0 \quad \forall v \in V_{\Gamma}. \end{cases} \quad (8)$$

Note that the only difference to the global eigenvalue problem (2) is that the functions u and v in (7) and (8) are elements of V_{Ω_i} or V_Γ instead of $H_0^1(\Omega)$. According to Theorem 1 it follows that (7) and (8) are the orthogonal projections of the global eigenvalue problem onto the subspaces $V_{\Omega_1}, V_{\Omega_2}$ and V_Γ .

The fixed-interface and the coupling mode eigenvalue problems each possess a countable family of eigensolutions which are given by

$$(\lambda_{i,j}, u_{i,j})_{j=1}^\infty \in \mathbb{R}_{>0} \times V_{\Omega_i} \quad \text{with } \lambda_{i,j} \leq \lambda_{i,j+1}$$

for the fixed-interface eigenvalue problem (7) and by

$$(\lambda_{\Gamma,j}, u_{\Gamma,j})_{j=1}^\infty \in \mathbb{R}_{>0} \times V_\Gamma \quad \text{with } \lambda_{\Gamma,j} \leq \lambda_{\Gamma,j+1}$$

for the coupling mode eigenvalue problem (8). Because the eigenfunctions $u_{i,k}$ and $u_{\Gamma,k}$ form a basis of $H_0^1(\Omega)$ we obtain

$$H_0^1(\Omega) = \bigcup_{i=1}^2 \text{span}\{u_{i,j} \mid j \in \mathbb{N}\} \cup \text{span}\{u_{\Gamma,j} \mid j \in \mathbb{N}\}. \quad (9)$$

In the third step the global eigenvalue problem is projected onto a suitable finite dimensional subspace of $H_0^1(\Omega)$ which is set up to represent the whole system. Selecting only these eigenfunctions in (9) which belong to a small eigenvalue this subspace is defined by

$$U_n := \bigcup_{i=1}^2 \text{span}\{u_{i,j} \mid j = 1, \dots, n_i\} \cup \text{span}\{u_{\Gamma,j} \mid j = 1, \dots, n_\Gamma\} \quad (10)$$

for given natural numbers n_1, n_2 and n_Γ with $n = n_1 + n_2 + n_\Gamma$.

The modal truncation performed in (10) is motivated by two reasons. One reason is that only the first $p(N) \ll N$ eigenfunctions can be well approximated by a finite element space using N *degrees of freedom* (short DOF), where $p(N)$ is for example $p(N) = N^{1/3}$ or $p(N) = N^{1/2}$. Another reason from a more practical point of view is that eigenfunctions belonging to larger eigenvalues have much lower participation in the response of the system than eigenfunctions belonging to lower eigenvalues.

Projecting the global eigenvalue problem (2) onto U_n the so called *reduced eigenvalue problem*

$$\begin{cases} \text{find } (\lambda, u) \in \mathbb{R} \times U_n \text{ such that} \\ a(u, v) = \lambda(u, v)_0 \quad \forall v \in U_n \end{cases}$$

is obtained with the eigensolutions

$$(\lambda_j^{(n)}, u_j^{(n)})_{j=1}^n \in \mathbb{R}_{>0} \times U_n \quad \text{with } \lambda_j^{(n)} \leq \lambda_{j+1}^{(n)}. \quad (11)$$

In the fourth and last step of AMLS the first n_{ev} eigensolutions (11) are computed which are approximating the sought eigensolutions $(\lambda_j, u_j)_{j=1}^{n_{ev}}$ of the global eigenvalue problem (2).

Theorem 2. *The coupling mode eigenvalue problem (8) is equivalent to the eigenvalue problem*

$$\begin{cases} \text{find } (\lambda, u) \in \mathbb{R} \times H_{00}^{1/2}(\Gamma) \text{ such that} \\ \langle \mathcal{S}u, v \rangle = \lambda \langle \mathcal{M}u, v \rangle \quad \forall v \in H_{00}^{1/2}(\Gamma) \end{cases} \quad (12)$$

where \mathcal{S} and \mathcal{M} are operators acting on the trace space $H_{00}^{1/2}(\Gamma)$ which are given in strong form by

$$\mathcal{S}\tau = \sum_{i=1}^2 ((A\nabla E_{\Omega_i}\tau) \cdot \mathbf{n}^i)|_{\Gamma} \quad \text{and} \quad \mathcal{M}\tau = \sum_{i=1}^2 -((A\nabla \mathcal{G}_i(E_{\Omega_i}\tau)) \cdot \mathbf{n}^i)|_{\Gamma}$$

for $\tau \in H_{00}^{1/2}(\Gamma)$. Here \mathbf{n}^i denotes the outward normal unit vector on Γ for the subdomain Ω_i ; E_{Ω_i} is the subdomain extension operator defined by $E_{\Omega_i}\tau := (E_{\Omega}\tau)|_{\Omega_i}$; and $\mathcal{G}_i(f)$ is the solution of the Dirichlet problem

$$\begin{cases} \text{find } \mathcal{G}_i(f) \in V_{\Omega_i} \text{ such that} \\ a(\mathcal{G}_i(f), v) = (f, v)_0 \quad \forall v \in V_{\Omega_i}, \end{cases} \quad (13)$$

i.e., \mathcal{G}_i is the Green's function of problem (13).

Proof. The proof of the theorem can be found in [6] in the context of an eigenvalue problem from linear elastodynamics, however, it can be applied for generic H^1 -elliptic bilinear forms. The operator \mathcal{M} is derived according to [6] and the operator \mathcal{S} according to [24]. \square

Remark 3. *i) \mathcal{S} is the so-called Steklov-Poincaré operator associated to the bilinear form $a(\cdot, \cdot)$ which is symmetric, continuous and coercive in $H_{00}^{1/2}(\Gamma)$ (cf. [24]). \mathcal{M} is the so-called mass operator associated to the bilinear form $a(\cdot, \cdot)$ (cf. [6]).*

ii) The fixed-interface eigenvalue problem (7) is equivalent to the eigenvalue problem

$$\begin{cases} \text{find } (\lambda, u) \in \mathbb{R} \times H_0^1(\Omega_i) \text{ such that} \\ a(u, v) = \lambda(u, v)_0 \quad \forall v \in H_0^1(\Omega_i). \end{cases} \quad (14)$$

iii) The benefit of the representation (14) and (12) compared to (7) and (8) is that the eigenvalue problems are solely solved and evaluated on the subdomains Ω_i respectively the interface Γ .

In this section we have seen how the global problem is projected onto orthogonal subspaces leading to three different subproblems defined on the subdomains Ω_i and the interface Γ . Eigenfunctions belonging to the smallest eigenvalues are selected from each of these subproblems to form a suitable finite dimensional subspace on which the global problem is projected. A reduced problem is obtained whose solutions are approximating the sought solutions of the global problem.

3.2 Single-Level Case: Algebraic Setting

The initial point of the description of AMLS in the algebraic setting is the discrete eigenvalue problem (4). For reasons of convenience the upper index of $\lambda^{(N)}$ and $x^{(N)}$ in (4) — indicating the DOF of the finite element discretisation — is left out in this particular section and the following eigenvalue problem

$$\begin{cases} \text{find } (\lambda, x) \in \mathbb{R} \times \mathbb{R}^N \text{ with} \\ Kx = \lambda Mx \end{cases} \quad (15)$$

is considered with the eigenpairs $(\lambda_j, x_j)_{j=1}^N \in \mathbb{R}_{>0} \times \mathbb{R}^N$ and $\lambda_j \leq \lambda_{j+1}$. Because the matrices K and M in (15) result from a finite element discretisation each row and column index is associated with a basis function which has typically a small support. Using the substructuring of $\Omega = \Omega_1 \cup \Omega_2$ with $\Gamma = \overline{\Omega}_1 \cap \overline{\Omega}_2$ from the section before, the row and column indices can be reordered in such a way that

$$K = \begin{matrix} & \Omega_1 & \Omega_2 & \Gamma \\ \Omega_1 & K_{11} & & K_{13} \\ \Omega_2 & & K_{22} & K_{23} \\ \Gamma & K_{31} & K_{32} & K_{33} \end{matrix} \quad \text{and} \quad M = \begin{matrix} & \Omega_1 & \Omega_2 & \Gamma \\ \Omega_1 & M_{11} & & M_{13} \\ \Omega_2 & & M_{22} & M_{23} \\ \Gamma & M_{31} & M_{32} & M_{33} \end{matrix} \quad (16)$$

holds with $K_{ij}, M_{ij} \in \mathbb{R}^{N_i \times N_j}$ and $N_1 + N_2 + N_3 = N$. The labels Ω_1, Ω_2 and Γ in (16) are indicating to which subset the indices are associated, i.e., if the supports of the corresponding basis functions are inside Ω_i or intersecting Γ .

Performing a block LDL^T -decomposition in the next step of AMLS the matrix K is block diagonalised by $K = L\tilde{K}L^T$ with $\tilde{K} = \text{diag}[K_{11}, K_{22}, \tilde{K}_{33}]$ and

$$L := \begin{bmatrix} \text{Id} & & \\ & \text{Id} & \\ K_{31}K_{11}^{-1} & K_{32}K_{22}^{-1} & \text{Id} \end{bmatrix}.$$

The submatrix \tilde{K}_{33} given by

$$\tilde{K}_{33} = K_{33} - K_{31}K_{11}^{-1}K_{13} - K_{32}K_{22}^{-1}K_{23}$$

is the *Schur complement* of $\text{diag}[K_1, K_2]$ in K and it is typically dense. The matrix M is transformed correspondingly by computing $\tilde{M} := L^{-1}ML^{-T}$ with

$$\tilde{M} = \begin{bmatrix} M_{11} & & \tilde{M}_{13} \\ & M_{22} & \tilde{M}_{23} \\ \tilde{M}_{31} & \tilde{M}_{32} & \tilde{M}_{33} \end{bmatrix}$$

where \tilde{M}_{33} is given by

$$\tilde{M}_{33} = M_{33} - \sum_{i=1}^2 \left(K_{3i}K_{ii}^{-1}M_{i3} + M_{3i}K_{ii}^{-1}K_{i3} - K_{3i}K_{ii}^{-1}M_{ii}K_{ii}^{-1}K_{i3} \right).$$

A part of the sparsity structure is lost in \widetilde{K} and \widetilde{M} . All submatrices \widetilde{K}_{ii} and \widetilde{M}_{ij} whose row or column indices are associated with the interface Γ are now typically dense.

The eigenvalue problems (K, M) and $(\widetilde{K}, \widetilde{M})$ are equivalent, i.e., the eigenvalues of both problems are equal and if \widetilde{x} is an eigenvector of $(\widetilde{K}, \widetilde{M})$ then $x = L^{-T}\widetilde{x}$ is an eigenvector of (K, M) .

At first glance, the reason for the applied eigenvalue problem transformation of (K, M) is not obvious. But it can be shown that the eigenvalue problem $(\widetilde{K}_{33}, \widetilde{M}_{33})$ corresponds to the finite element discretisation of the continuous coupling mode eigenvalue problem (12), cf. [24] and [6], and the eigenvalue problems (K_{ii}, M_{ii}) for $i = 1, 2$ are the respective finite element discretisations of the continuous fixed-interface eigenvalue problems (14).

In the next step of the AMLS method the eigenvalue problems (K_{11}, M_{11}) , (K_{22}, M_{22}) and $(\widetilde{K}_{33}, \widetilde{M}_{33})$ are solved. Corresponding to the modal truncation applied in (10) in the continuous setting, only those eigenpairs of (K_{11}, M_{11}) , (K_{22}, M_{22}) and $(\widetilde{K}_{33}, \widetilde{M}_{33})$ are computed which belong to the smallest $k_i \in \mathbb{N}$ eigenvalues for given $k_i \leq N_i$ and $i = 1, 2, 3$. In the following these partial eigensolutions are

$$K_{ii} S_i = M_{ii} S_i D_i \quad (i = 1, 2) \quad \text{and} \quad \widetilde{K}_{33} S_3 = \widetilde{M}_{33} S_3 D_3 \quad (17)$$

where the matrix $S_i \in \mathbb{R}^{N_i \times k_i}$ contains column-wise the k_i selected eigenvectors and the diagonal matrix $D_i \in \mathbb{R}^{k_i \times k_i}$ contains the corresponding eigenvalues ($i = 1, 2, 3$). Furthermore, the eigenvectors are normalised by $S_i^T M_{ii} S_i = \text{Id}$ ($i = 1, 2$) and $S_3^T \widetilde{M}_{33} S_3 = \text{Id}$.

Remark 4. *How many eigenvectors have to be selected in (17) from each subproblem is not easy to answer. On one side k_i should be small to obtain in the further proceeding of AMLS a reduced problem of small size which can be easily solved. On the other side enough spectral information has to be kept to obtain sufficiently good eigenpair approximations from the reduced problem. Using different approaches in [11] and [27] heuristics have been derived on how to select eigenpairs. These heuristics are based purely on the analysis of the algebraic eigenvalue problem $(\widetilde{K}, \widetilde{M})$ without using any geometry information of the underlying partial differential equation. One possible strategy for the eigenpair selection in (17) — which is also used in Section 7 where numerical results are presented — is as follows: Select in each subproblem only those eigenpairs whose eigenvalues are smaller than a given truncation bound $\omega > 0$.*

In the next step of AMLS the block diagonal matrix

$$S := \text{diag}[S_1, S_2, S_3] \in \mathbb{R}^{N \times k}$$

with $k := k_1 + k_2 + k_3 \ll N$ is defined which is built of all selected eigenvectors. Then the matrices $\widehat{K} := S^T \widetilde{K} S \in \mathbb{R}^{k \times k}$ and $\widehat{M} := S^T \widetilde{M} S \in \mathbb{R}^{k \times k}$ are computed where it holds

$$\widehat{K} = \text{diag}[D_1, D_2, D_3] \quad \text{and} \quad \widehat{M} = \begin{bmatrix} \text{Id} & & \widehat{M}_{13} \\ & \text{Id} & \widehat{M}_{23} \\ \widehat{M}_{31} & \widehat{M}_{32} & \text{Id} \end{bmatrix},$$

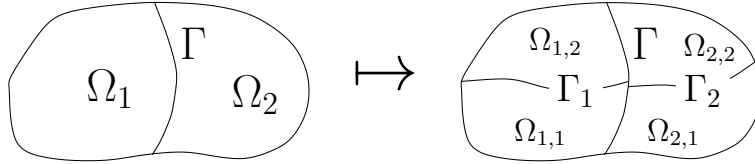


Figure 2: Extending the single-level substructuring of Ω to a two-level substructuring.

and a *reduced eigenvalue problem* is obtained

$$\begin{cases} \text{find } (\hat{\lambda}, \hat{x}) \in \mathbb{R} \times \mathbb{R}^k \text{ with} \\ \hat{K} \hat{x} = \hat{\lambda} \hat{M} \hat{x} \end{cases} \quad (18)$$

with the eigenpairs $(\hat{\lambda}_j, \hat{x}_j)_{j=1}^k \in \mathbb{R}_{>0} \times \mathbb{R}^k$ and $\hat{\lambda}_j \leq \hat{\lambda}_{j+1}$.

In the last step of AMLS the smallest n_{ev} eigenpairs of (18) are computed. The vectors

$$\hat{y}_j := L^{-T} S \hat{x}_j \quad \text{with } j = 1, \dots, k \quad (19)$$

are Ritz-vectors of the original eigenvalue problem (K, M) respective to the subspace spanned by the columns of the matrix S , and $\hat{\lambda}_j$ are the respective Ritz-values. The pairs $(\hat{\lambda}_j, \hat{y}_j)_{j=1}^{n_{ev}}$ are approximating the sought smallest n_{ev} eigenpairs of the original eigenvalue problem (K, M) .

Remark 5. *i) Because the eigenpairs of the reduced eigenvalue problem (\hat{K}, \hat{M}) are primarily used to approximate the eigensolutions of the continuous problem (2) and not the eigenpairs of the discretised problem (K, M) , the approximation error of AMLS is influenced by the finite element discretisation and the modal truncation applied in (17). As long as the error caused by the modal truncation is of the same order as the discretisation error the eigenpair approximations derived from the reduced problem (\hat{K}, \hat{M}) are of comparable quality as the eigenpair approximations derived from the problem (K, M) .*

ii) The reduced eigenvalue problem (\hat{K}, \hat{M}) is much easier to solve than the original eigenvalue problem (K, M) because the number of selected eigenpairs in (17) is typically quite small and therefore the order of the reduced problem is much smaller than the order of the original problem. Furthermore, the matrices \hat{K} and \hat{M} are sparse, i.e., \hat{M} has the same block zero structure as M and \hat{K} is even a diagonal matrix. Correspondingly a sparse eigensolver can be used to compute the smallest n_{ev} eigenpairs of (\hat{K}, \hat{M}) .

iii) The matrix partitioning (16) can also be done in a purely algebraic way by applying graph partitioning algorithms like nested dissection to the graph of the matrix $|K| + |M|$.

3.3 Multi-Level Case: Algebraic Setting

The single-level version of the AMLS method explained in the section before can be easily extended to the multi-level case. Using the substructuring from the single-level case we further subdivide the subdomains Ω_1 and Ω_2 each into two non-overlapping subdomains which share some interface as it is illustrated in Figure 2. This substructuring can be applied

again recursively to the resulting subdomains until a certain level is exceeded or the size of the subdomains falls below some given limit.

The further proceeding of AMLS in the multi-level case is analogous to the single-level case. As in (16) the row and column indices of the matrices K and M are reordered to achieve a matrix partitioning according to the performed domain substructuring. For example the matrix partitioning of K corresponding to the domain substructuring applied in Figure 2 is

$$K = \begin{array}{c} \Omega_{1,1} \\ \Omega_{1,2} \\ \Gamma_1 \\ \Omega_{2,1} \\ \Omega_{2,2} \\ \Gamma_2 \\ \Gamma \end{array} \begin{bmatrix} \Omega_{1,1} & \Omega_{1,2} & \Gamma_1 & \Omega_{2,1} & \Omega_{2,2} & \Gamma_2 & \Gamma \\ K_{11} & & K_{13} & & & & K_{17} \\ & K_{22} & K_{23} & & & & K_{27} \\ K_{31} & K_{32} & K_{33} & & & & K_{37} \\ & & & K_{44} & & K_{46} & K_{47} \\ & & & & K_{55} & K_{56} & K_{57} \\ & & & K_{64} & K_{65} & K_{66} & K_{67} \\ K_{71} & K_{72} & K_{73} & K_{74} & K_{75} & K_{76} & K_{77} \end{bmatrix}. \quad (20)$$

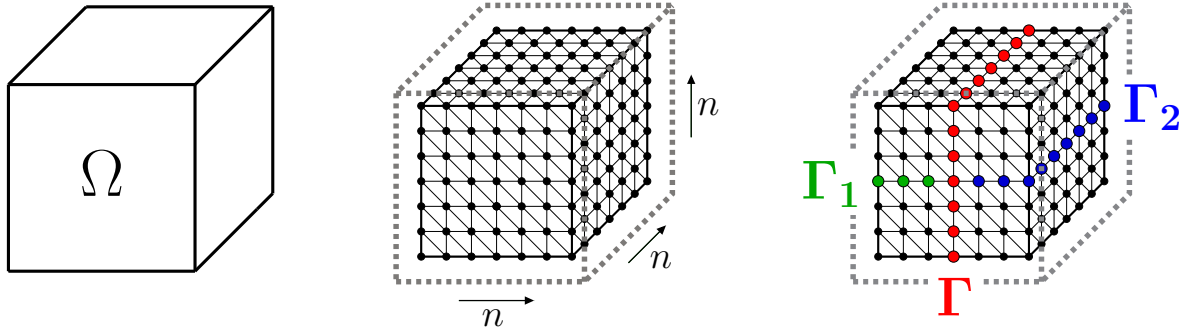
K_{ij} is the submatrix of K in block row i and block column j with $i, j = 1, \dots, m$ where m is equal to the number of subdomains and interfaces contained in Ω . In the next step the eigenvalue problem (K, M) is transformed equivalently to (\tilde{K}, \tilde{M}) , i.e., K is block diagonalised via $K = L\tilde{K}L^T$ by performing a block LDL^T -decomposition and M is transformed correspondingly by $\tilde{M} = L^{-1}ML^{-T}$. Due to the transformation a part of the sparsity structure is lost in \tilde{K} and \tilde{M} . All submatrices \tilde{K}_{ii} and \tilde{M}_{ij} are now typically dense if their respective row or column indices are associated with an interface. In the next step the partial eigensolutions of the subproblems $(\tilde{K}_{ii}, \tilde{M}_{ii})$ are computed. Note that $\tilde{K}_{ii} = K_{ii}$ and $\tilde{M}_{ii} = M_{ii}$ if their row indices are associated with one of the subdomains. Let the partial eigensolution be given again by

$$\tilde{K}_{ii} S_i = \tilde{M}_{ii} S_i D_i \quad \text{with} \quad S_i^T \tilde{M}_{ii} S_i = \text{Id}$$

for $i = 1, \dots, m$, where $S_i \in \mathbb{R}^{N_i \times k_i}$ contains column-wise $k_i \leq N_i$ selected eigenvectors and the diagonal matrix $D_i \in \mathbb{R}^{k_i \times k_i}$ the corresponding eigenvalues. In the next step the reduced eigenvalue problem (\hat{K}, \hat{M}) is obtained by computing $\hat{K} := S^T K S$ and $\hat{M} := S^T M S$ with $S := \text{diag}[S_1, \dots, S_m]$. Finally, the n_{ev} smallest eigenpairs of the reduced eigenvalue problem are computed where eigenpair approximations of the original eigenvalue problem (K, M) are obtained by (19).

For further illustration we refer to [12] where a two-level version of AMLS in the algebraic setting is described, and for the description of the multi-level case in the continuous setting we refer to [6]. To summarise the AMLS method an overview of all necessary operations is given in Table 1.

The benefit of the multi-level approach is that the substructuring of the domain or respectively the partitioning of the matrices K and M can be applied recursively until eventually in (17) the size of the subproblems $(\tilde{K}_{ii}, \tilde{M}_{ii})$ is small enough to be solved easily. However, the recursion will only affect the size of subproblems corresponding to subdomains and not those related to interfaces. When the spatial domain Ω is three-dimensional this is a bottleneck.



(a) Domain $\Omega = (0, 1)^3$. (b) Discretised domain with n^3 DOF; DOF are indicated by small dots and only the grid associated to the DOF is marked. (c) Two-level substructuring of the discretised domain leading to interfaces of the size $\mathcal{O}(n^2)$.

Figure 3: Discretisation of the domain $\Omega = (0, 1)^3$ and performing a two-level substructuring.

4 Efficiency Problems in the Three-Dimensional Case

In the following we refer to submatrices whose row or column indices are associated with an interface as *interface matrices*. In the three-dimensional case these interface matrices are getting relatively large in AMLS which leads to very high computational costs. In contrast to submatrices which are associated only with subdomains the size of the interface matrices cannot be reduced by further substructuring as discussed in the previous section.

To illustrate this we take a look at the initial eigenvalue problem (2) with the domain $\Omega = (0, 1)^3$. To solve the problem with AMLS it has to be discretised first. This can be done for example by decomposing $\Omega = (0, 1)^3$ into $n + 1$ equispaced subintervals in each direction and using standard P1 finite elements, cf. Figure 3b. The resulting eigenvalue problem is given in (4) where the matrices K and M are of size $N \times N$ with $N = n^3$. Assuming that a two-level substructuring is used in AMLS we obtain a matrix partitioning like in (20). The number of rows or columns of the interface matrices are $\mathcal{O}(N^{2/3})$ as it is illustrated in Figure 3c. These interface matrices are relatively large and their size cannot be reduced by further substructuring.

During the AMLS method a couple of matrix operations have to be performed on these interface matrices, e.g., computing the inverse, the matrix product or the partial eigen-solution. Beside that the interface matrices are relatively large they are dense as well. For example in the two-level version of AMLS the inverse of the interface matrices \tilde{K}_{33} and \tilde{K}_{66} has to be computed when the block LDL^T -decomposition is performed to block diagonalise $K = L\tilde{K}L^T$. These operations alone lead to costs of $\mathcal{O}((N^{2/3})^3) = \mathcal{O}(N^2)$. A way to help out here are so-called hierarchical matrices which are shortly introduced in the next section.

5 Hierarchical matrices

\mathcal{H} -matrices [18, 19] are data-sparse but possibly dense matrices. The underlying idea is to reorder the row and columns of a matrix such that certain submatrices can be represented or approximated by low rank matrices. To represent such a fully populated but data-sparse matrix of size $N \times N$ only $\mathcal{O}(N \log^2 N)$ data is necessary instead of storing N^2 entries. Moreover, \mathcal{H} -matrices provide exact matrix-vector multiplication and approximated matrix(-matrix) operations (e.g. multiplication, addition, inversion, LU -factorisation) which are performed in almost linear complexity $\mathcal{O}(N \log^\alpha N)$ with $\alpha = 1, \dots, 4$ (cf. [13, 15]).

The stiffness matrix resulting from the finite element discretisation of an elliptic partial differential operator is sparse. However, its inverse and its LU -factors are fully populated. In [3] and [14] it is shown that the inverse and the LU -factors can be approximated by \mathcal{H} -matrices and that these approximations can be computed with almost linear complexity. This motivates to use the fast \mathcal{H} -matrix algebra in the AMLS method to compute the block diagonalisation $K = L\tilde{K}L^T$ and the matrix transformation $\tilde{M} = L^{-1}ML^{-T}$.

To do this the sparse matrices K and M have to be converted into \mathcal{H} -matrices. For this purpose a suitable \mathcal{H} -matrix format has to be provided which is based on the geometry data of the underlying partial differential equation. To introduce this \mathcal{H} -matrix format and the basic concept of \mathcal{H} -matrices we first explain how the inverse of a stiffness matrix is approximated by an \mathcal{H} -matrix.

Assume $A \in \mathbb{R}^{N \times N}$ is the stiffness matrix resulting from the finite element discretisation of an elliptic partial differential operator. The matrix A is sparse, however, its inverse A^{-1} is fully populated. Recalling the definition of the stiffness matrix in (5) each row and column index $i \in I := \{1, \dots, N\}$ of A and respectively of A^{-1} is associated with a basis function $\varphi_i^{(N)}$ of the underlying finite element space V_N . For each index set $t \subset I$ we define its support by

$$\Omega_t := \bigcup_{i \in t} \text{supp}(\varphi_i^{(N)}).$$

Correspondingly each submatrix

$$A^{-1}|_{s \times t} := ((A^{-1})_{ij})_{i \in s, j \in t} \quad \text{with } s, t \subset I$$

of A^{-1} is associated with geometry information. Based on the geometric separation of the index sets s and t certain subblocks $s \times t \subset I \times I$ can be identified that allow a low rank approximation of the respective submatrices $A^{-1}|_{s \times t}$. More precisely, submatrices $A^{-1}|_{s \times t}$ whose index sets s and t fulfil the so-called *admissibility condition*

$$\min\{\text{diam}(\Omega_s), \text{diam}(\Omega_t)\} \leq \eta \text{dist}(\Omega_s, \Omega_t) \quad (21)$$

are well suited for a low rank approximation (cf. [3]). The parameter $\eta > 0$ controls the number of admissible subblocks $s \times t$ and is set to $\eta := 50$ in the rest of the paper. The quantities

$$\begin{aligned} \text{diam}(\Omega_t) &:= \max\{\|x - y\|_2 \mid x, y \in \Omega_t\} \quad \text{and} \\ \text{dist}(\Omega_s, \Omega_t) &:= \min\{\|x - y\|_2 \mid x \in \Omega_s, y \in \Omega_t\} \end{aligned}$$

are the diameter and the distance of the supports of s and t . Subblocks $s \times t$ fulfilling the admissibility condition (21) are called *admissible* and the corresponding submatrices $A|_{s \times t}$ are approximated by so-called *Rk*-matrices which are defined as follows.

Definition 6 (*Rk*-matrix). *Let be $k, m, n \in \mathbb{N}_0$. The matrix $R \in \mathbb{R}^{n \times m}$ is called *Rk*-matrix if it is factorised by*

$$R = UV^T \text{ for suitable matrices } U \in \mathbb{R}^{n \times k}, V \in \mathbb{R}^{m \times k}.$$

When the rank k is small compared to n and m the representation of an *Rk*-matrix $R \in \mathbb{R}^{n \times m}$ is much cheaper than in full-matrix representation because only $k(n + m)$ entries have to be stored instead of nm . Furthermore, the product and the sum of two *Rk*-matrices can be evaluated much more efficiently than in full-matrix representation when k is small.

To exploit the low rank approximation property of submatrices fulfilling (21) we reorder the row and column indices of A^{-1} . For this purpose the index set I is divided according to a geometric bisection of its support into two disjoint index sets $s, t \in I$ with $I = s \dot{\cup} t$. In this context s and t are denoted as *sons* of I and $S(I) := \{s, t\}$ as the *set of sons* of I . This geometric bisection is applied recursively to the son index sets until the cardinality of an index set falls below some given limit $n_{min} \in \mathbb{N}$. Such a partitioning is illustrated in Figure 4a and 4b for a two-dimensional problem. The described geometric bisection results in a disjoint partition of the index set I where the obtained subsets of the partitioning tend to be geometrically separated.

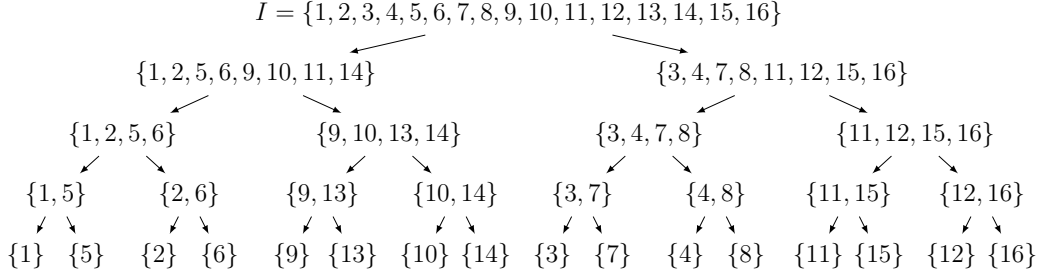
Given the admissibility condition (21) and the partitioning of the index set I the \mathcal{H} -matrix format of A^{-1} is constructed by applying algorithm 1 to $I \times I$. Using this algorithm $I \times I$ is recursively subdivided into subblocks $s \times t$ until the subblock gets admissible or the size of the subblock falls below the limit n_{min} as it is illustrated in Figure 4c. Submatrices $A^{-1}|_{s \times t}$ of admissible blocks $s \times t$ are represented in the *Rk*-matrix format and submatrices of inadmissible blocks are represented in the full matrix format. To control the approximation quality of the *Rk*-matrix approximation the fixed rank is replaced by an adaptive rank. For a desired approximation accuracy $\varepsilon > 0$ each submatrix $A^{-1}|_{s \times t}$ corresponding to an admissible subblock $s \times t$ can be approximated by an *Rk*-matrix R such that

$$\frac{\|A^{-1}|_{s \times t} - R\|_2}{\|A^{-1}|_{s \times t}\|_2} \leq \varepsilon \tag{22}$$

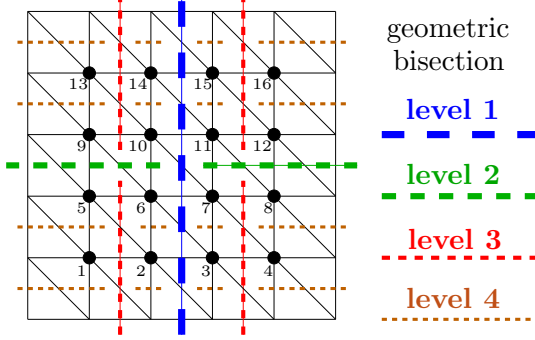
where the rank $k \in \mathbb{N}_0$ is as small as possible (cf. [13]).

Let $(A^{\mathcal{H}})^{-1}$ denote the \mathcal{H} -matrix approximation of A^{-1} computed by Algorithm 1. The error $\|A^{-1} - (A^{\mathcal{H}})^{-1}\|$ is influenced by the chosen accuracy ε in (22). The matrix A^{-1} is fully populated, however, for its \mathcal{H} -matrix approximation only $\mathcal{O}(N \log^2 N)$ data is needed for storage. Furthermore, $(A^{\mathcal{H}})^{-1}$ can be computed in $\mathcal{O}(N \log^\alpha N)$, cf. [13], using a recursive algorithm applied block-wise to the matrix structure and exploiting the *Rk*-matrix representation of submatrices fulfilling (21).

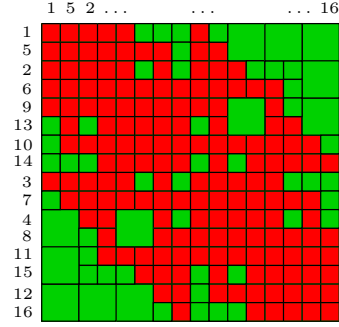
To compute the block diagonalisation $K = L\tilde{K}L^T$ and the matrix transformation $\tilde{M} = L^{-1}ML^{-T}$ by the fast \mathcal{H} -matrix algebra we slightly change the described \mathcal{H} -matrix format, cf. [15]. First we apply a nested dissection as in the classical AMLS method, i.e., the domain



(a) Disjoint partitioning of the index set I corresponding to the applied geometric bisection.



(b) Geometric bisection of the domain $\Omega = (0,1)^2$ using $n_{min} = 1$. The indices $i \in I = \{1, \dots, 16\}$ of the nodal points of the basis functions are numbered from 1 in the lower left to 16 in the upper right corner.



(c) \mathcal{H} -Matrix structure of $A^{-1} \in \mathbb{R}^{16 \times 16}$ according to the applied partitioning of I using admissibility condition (21) and $n_{min} = 1$; admissible blocks are coloured green, inadmissible red.

Figure 4: \mathcal{H} -matrix construction for the inverse of the stiffness matrix resulting from a finite element discretisation of an elliptic partial differential operator on $\Omega = (0,1)^2$ using standard P1 finite elements on an equispaced grid with 16 DOF.

Ω is recursively partitioned into several subdomains which are divided by interfaces. The row and column indices of K and M are reordered according to the performed partitioning of Ω and a matrix partitioning, e.g., of the form (16) or (20) is obtained. As discussed in Section 3 some of the submatrices \tilde{K}_{ij} and \tilde{M}_{ij} are fully populated, however, they can be approximated by \mathcal{H} -matrices. For this purpose we apply additionally a geometric bisection to the index sets associated with the subdomains and interfaces, and reorder the row and column indices of the submatrices \tilde{K}_{ij} and \tilde{M}_{ij} correspondingly. In Figure 6 the described domain partitioning is illustrated.

Using the \mathcal{H} -matrix format resulting from the matrix partitioning described above, the block diagonalisation of K and the transformation of M can be computed by the fast \mathcal{H} -matrix algebra by

$$K \approx L^{\mathcal{H}} \tilde{K}^{\mathcal{H}} (L^{\mathcal{H}})^T \quad \text{and} \quad \tilde{M}^{\mathcal{H}} \approx (L^{\mathcal{H}})^{-1} M (L^{\mathcal{H}})^{-T}. \quad (23)$$

An example of the resulting matrix structure is given in Figure 7 for the matrix $\tilde{M}^{\mathcal{H}}$. The \mathcal{H} -matrix operations in (23) are performed not exactly but only approximatively. The resulting

Algorithm 1 \mathcal{H} -Matrix Construction

```

procedure CONSTRUCTHMATRIX( $A^{-1}$ ,  $\varepsilon$ ,  $n_{min}$ ,  $s \times t$ )
  if  $s \times t$  is admissible then
    approximate  $A^{-1}|_{s \times t}$  by Rk-matrix with accuracy  $\varepsilon$ ;
  else if  $\min\{\#s, \#t\} \leq n_{min}$  then
    represent  $A^{-1}|_{s \times t}$  by a full matrix;  $\triangleright n_{min}$  affects the minimal size of the submatrices
  else
     $S(s \times t) := \{s' \times t' \mid s' \in S(s), t' \in S(t)\}$ ;  $\triangleright S(t)$  denotes the set of sons of  $t \subset I$ 
    for all  $s' \times t' \in S(s \times t)$  do
      CONSTRUCTHMATRIX( $A^{-1}$ ,  $\varepsilon$ ,  $n_{min}$ ,  $s' \times t'$ );
    end for
  end if
end procedure
  
```

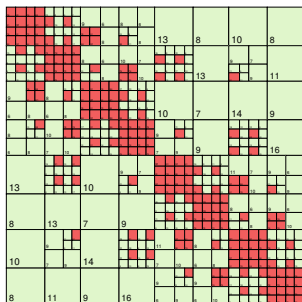


Figure 5: \mathcal{H} -matrix structure of $A^{-1} \in \mathbb{R}^{N \times N}$ with $N = 2500$ using admissible condition (21) and $n_{min} = 100$ where A is the stiffness matrix resulting from a finite element discretisation of an elliptic partial differential operator on $\Omega = (0, 1)^2$.

approximation errors $\|L - L^{\mathcal{H}}\|_2$, $\|\tilde{K} - \tilde{K}^{\mathcal{H}}\|_2$ and $\|\tilde{M} - \tilde{M}^{\mathcal{H}}\|_2$ are influenced by the chosen accuracy ε in (22).

6 AMLS combined with \mathcal{H} -matrices

In this section a more refined version of the AMLS method using the fast \mathcal{H} -matrix algebra is presented. The benefit of the use of the \mathcal{H} -matrices is a reduction in computational time and storage requirements. However, an additional error due the use of \mathcal{H} -matrices occurs which can influence the quality of the computed eigenpair approximations. This problem is discussed in the following but first the new method, called \mathcal{H} -AMLS, is introduced.

As in the classical AMLS method in the first step of \mathcal{H} -AMLS a nested dissection is applied. To use the fast \mathcal{H} -matrix algebra additionally a geometric bisection is performed as described in the previous section. In the next step we compute as in (23) the block diagonalisation of K and the corresponding matrix transformation of M using the fast \mathcal{H} -matrix algebra.

The further proceeding of \mathcal{H} -AMLS is analogous to the classical AMLS method. Submatrices of $\tilde{K}^{\mathcal{H}}$ and $\tilde{M}^{\mathcal{H}}$ according to block row i and block column j are denoted by $\tilde{K}_{ij}^{\mathcal{H}}$ and $\tilde{M}_{ij}^{\mathcal{H}}$. In the next step the partial eigensolutions of the subproblems $(\tilde{K}_{ii}^{\mathcal{H}}, \tilde{M}_{ii}^{\mathcal{H}})$ are computed for $i = 1, \dots, m$ which are given by

$$\tilde{K}_{ii}^{\mathcal{H}} \mathbf{S}_i = \tilde{M}_{ii}^{\mathcal{H}} \mathbf{S}_i \mathbf{D}_i \quad \text{with} \quad \mathbf{S}_i^T \tilde{M}_{ii}^{\mathcal{H}} \mathbf{S}_i = \text{Id}, \quad (24)$$

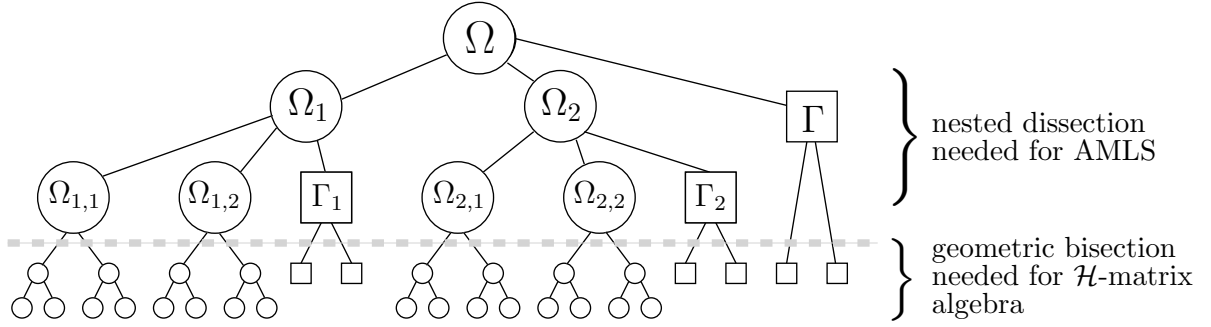


Figure 6: Schematic example of the partitioning of the domain Ω applied in \mathcal{H} -AMLS: A two-level nested dissection (necessary for AMLS, cf. Figure 2) is applied followed by an additional two-level geometric bisection of the subdomains and a one-level geometric bisection of the interfaces (necessary for \mathcal{H} -matrix approximation).

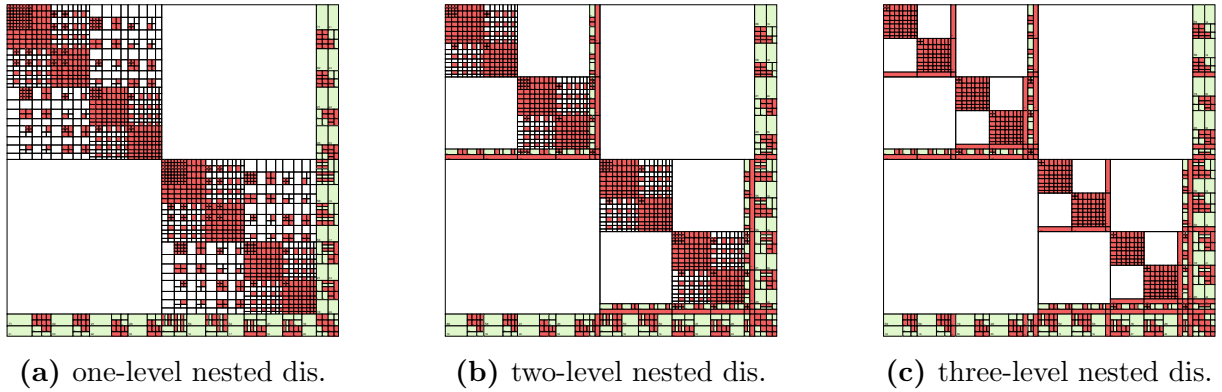


Figure 7: \mathcal{H} -matrix structure of $\widetilde{M}^{\mathcal{H}}$ using a one, two and three-level nested dissection. Red blocks represent full matrices, green blocks Rk -matrices and white blocks submatrices equal to zero which don't cause computational costs in the \mathcal{H} -matrix algebra.

where the matrix $\mathbf{S}_i \in \mathbb{R}^{N_i \times k_i}$ contains column-wise $k_i \leq N_i$ selected eigenvectors and the diagonal matrix $\mathbf{D}_i \in \mathbb{R}^{k_i \times k_i}$ contains the corresponding eigenvalues. Because in general the matrices $\widetilde{K}_{ii}^{\mathcal{H}}$ and $\widetilde{M}_{ii}^{\mathcal{H}}$ slightly differ from \widetilde{K}_{ii} and \widetilde{M}_{ii} the corresponding eigensolutions (17) and (24) can differ as well. To indicate this difference in the \mathcal{H} -AMLS method bold symbols are used for the corresponding matrices and symbols.

In the next step we define $\mathbf{S} := \text{diag}[\mathbf{S}_1, \dots, \mathbf{S}_m]$ and the matrices

$$\widehat{\mathbf{K}} := \mathbf{S}^T \widetilde{\mathbf{K}}^{\mathcal{H}} \mathbf{S} \in \mathbb{R}^{k \times k} \quad \text{and} \quad \widehat{\mathbf{M}} := \mathbf{S}^T \widetilde{\mathbf{M}}^{\mathcal{H}} \mathbf{S} \in \mathbb{R}^{k \times k}$$

are computed which lead to the so-called \mathcal{H} -reduced eigenvalue problem

$$\begin{cases} \text{find } (\widehat{\boldsymbol{\lambda}}, \widehat{\mathbf{x}}) \in \mathbb{R} \times \mathbb{R}^k \text{ with} \\ \widehat{\mathbf{K}} \widehat{\mathbf{x}} = \widehat{\boldsymbol{\lambda}} \widehat{\mathbf{M}} \widehat{\mathbf{x}} \end{cases} \quad (25)$$

where the eigenpairs be given by $(\widehat{\boldsymbol{\lambda}}_j, \widehat{\mathbf{x}}_j)_{j=1}^k \in \mathbb{R}_{>0} \times \mathbb{R}^k$ with $\widehat{\boldsymbol{\lambda}}_j \leq \widehat{\boldsymbol{\lambda}}_{j+1}$. In the last step the smallest n_{ev} eigenpairs of (25) are computed leading to the eigenpair approximations

$(\widehat{\lambda}_j, \widehat{\mathbf{y}}_j)_{j=1}^{n_{ev}}$ of the original problem (K, M) with $\widehat{\mathbf{y}}_j := (L^{\mathcal{H}})^{-T} \mathbf{S} \widehat{\mathbf{x}}_j$. In contrast to the classical AMLS method, in general $\widehat{\lambda}_j$ is not equal to the Rayleigh quotient

$$\widehat{\lambda}_j^{rq} := \widehat{\mathbf{y}}_j^T K \widehat{\mathbf{y}}_j / \widehat{\mathbf{y}}_j^T M \widehat{\mathbf{y}}_j \quad (26)$$

since the matrix operations in (23) are performed only approximatively. Typically the Rayleigh quotients $\widehat{\lambda}_j^{rq}$ deliver better approximations of the sought eigenvalues λ_j than $\widehat{\lambda}_j$, especially when the chosen accuracy ε of \mathcal{H} -matrix approximation is coarse. To compare the classical AMLS method with the new \mathcal{H} -AMLS method an overview of both methods is given in Table 1 where the different tasks of the methods are referred to **(T1)**–**(T8)**.

Task	Matrix Operations AMLS	Matrix Operations \mathcal{H} -AMLS
(T1) partition matrices K and M	nested dissection reordering, cf. (16) and (20)	nested dissection reordering, cf. (16) and (20), with subsequent geometric bisection (cf. Section 5)
(T2) block diagonalise K	$K = L\widetilde{K}L^T \rightarrow$ expensive because of large-sized, dense interface matrices	$K \approx L^{\mathcal{H}}\widetilde{K}^{\mathcal{H}}(L^{\mathcal{H}})^T \rightarrow$ using fast \mathcal{H}-matrix algebra done in $\mathcal{O}(N \log^\alpha N)$
(T3) transform M	$\widetilde{M} = L^{-1}ML^{-T} \rightarrow$ expensive because of large-sized, dense interface matrices	$\widetilde{M}^{\mathcal{H}} \approx (L^{\mathcal{H}})^{-1}M(L^{\mathcal{H}})^{-T} \rightarrow$ using fast \mathcal{H}-matrix algebra done in $\mathcal{O}(N \log^\alpha N)$
(T4) compute partial eigensolution for $i = 1, \dots, m$	$\widetilde{K}_{ii}S_i = \widetilde{M}_{ii}S_iD_i \rightarrow$ expensive when \widetilde{K}_{ii} and \widetilde{M}_{ii} are interface matrices because they are dense	$\widetilde{K}_{ii}^{\mathcal{H}}S_i = \widetilde{M}_{ii}^{\mathcal{H}}S_iD_i \rightarrow$ use fast \mathcal{H}-matrix algebra when $\widetilde{K}_{ii}^{\mathcal{H}}$ and $\widetilde{M}_{ii}^{\mathcal{H}}$ are interface matrices
(T5) define subspace	$S := \text{diag}(S_1, \dots, S_m) \in \mathbb{R}^{N \times k}$ with $k = \sum_{i=1}^m k_i$	$\mathbf{S} := \text{diag}(\mathbf{S}_1, \dots, \mathbf{S}_m) \in \mathbb{R}^{N \times k}$ with $k = \sum_{i=1}^m k_i$
(T6) compute reduced eigenvalue problem	$\widehat{K} := S^T \widetilde{K} S \in \mathbb{R}^{k \times k}$, $\widehat{M} := S^T \widetilde{M} S \in \mathbb{R}^{k \times k}$	$\widehat{\mathbf{K}} := \mathbf{S}^T \widetilde{K}^{\mathcal{H}} \mathbf{S} \in \mathbb{R}^{k \times k}$, $\widehat{\mathbf{M}} := \mathbf{S}^T \widetilde{M}^{\mathcal{H}} \mathbf{S} \in \mathbb{R}^{k \times k} \rightarrow$ use fast \mathcal{H}-matrix algebra for computation
(T7) solve reduced eigenvalue problem	$\widehat{K} \widehat{\mathbf{x}}_j = \widehat{\lambda}_j \widehat{M} \widehat{\mathbf{x}}_j \quad \text{for } j = 1, \dots, n_{ev}$	$\widehat{\mathbf{K}} \widehat{\mathbf{x}}_j = \widehat{\lambda}_j \widehat{\mathbf{M}} \widehat{\mathbf{x}}_j \quad \text{for } j = 1, \dots, n_{ev}$
(T8) transformation of eigenvectors	$\widehat{\mathbf{y}}_j := L^{-T} S \widehat{\mathbf{x}}_j \quad \text{for } j = 1, \dots, n_{ev}$	$\widehat{\mathbf{y}}_j := (L^{\mathcal{H}})^{-T} \mathbf{S} \widehat{\mathbf{x}}_j \quad \text{for } j = 1, \dots, n_{ev}$
final eigenpair approximations	$(\widehat{\lambda}_j, \widehat{\mathbf{y}}_j) \quad \text{for } j = 1, \dots, n_{ev}$	$(\widehat{\lambda}_j^{rq}, \widehat{\mathbf{y}}_j) \quad \text{for } j = 1, \dots, n_{ev}$ with $\widehat{\lambda}_j^{rq} := \widehat{\mathbf{y}}_j^T K \widehat{\mathbf{y}}_j / \widehat{\mathbf{y}}_j^T M \widehat{\mathbf{y}}_j$

Table 1: Overview of the classical AMLS and the new \mathcal{H} -AMLS method.

6.1 Computational Costs

Beside N and the number of sought eigenpairs n_{ev} the computational costs of \mathcal{H} -AMLS depend on the chosen accuracy ε of the \mathcal{H} -matrix operation in (23) and the number of

selected eigenvectors k_i in (24), or respectively, the chosen truncation bound ω (cf. Remark 4). A coarser accuracy ε and a smaller truncation bound ω result in faster computations and reduced memory requirements of \mathcal{H} -AMLS. Of course these parameters can be chosen arbitrarily, however, their choice depends on the desired approximation accuracy of the sought n_{ev} eigenpairs. This issue is discussed in the next paragraph and in Section 7.

Comparing the different tasks it can be seen that the \mathcal{H} -AMLS method is much faster than the classical AMLS method. The computational costs of task **(T1)** are negligible. The computational costs for task **(T2)** and **(T3)** are of the order $\mathcal{O}(N \log^\alpha N)$ in \mathcal{H} -AMLS whereas in classical AMLS they are at least of the order $\mathcal{O}(N^2)$ in the three-dimensional case (cf. Section 4). Also the computation of the partial eigensolutions (task **(T4)**) are faster in the \mathcal{H} -AMLS method. The submatrices $\widetilde{K}_{ii}^{\mathcal{H}}$ and $\widetilde{M}_{ii}^{\mathcal{H}}$ whose row and column indices are associated to an interface are data-sparse \mathcal{H} -matrices and not unstructured dense matrices as assumed in the classical AMLS method. Correspondingly an eigensolver exploiting the \mathcal{H} -structure can be applied in (24) instead of an eigensolver for dense matrices as it is done in classical AMLS. For example the eigensolver \mathcal{H} -SIS using the fast \mathcal{H} -matrix algebra, which is presented in Section 7, can be applied. Also the \mathcal{H} -matrix structure of $\widetilde{K}^{\mathcal{H}}$ and $\widetilde{M}^{\mathcal{H}}$ can be exploited in \mathcal{H} -AMLS using the fast \mathcal{H} -matrix-vector multiplication for the computation of $\widehat{\mathbf{K}}$ and $\widehat{\mathbf{M}}$. Correspondingly the computation of task **(T6)** is in \mathcal{H} -AMLS faster than in classical AMLS. Only the computational costs of task **(T7)** are the same in both methods. The reduced eigenvalue problems $(\widehat{K}, \widehat{M})$ and $(\widehat{\mathbf{K}}, \widehat{\mathbf{M}})$ are both sparse and correspondingly a sparse eigensolver like ARPACK can be used to compute the smallest n_{ev} eigenpairs (cf. Remark 5ii)). Finally in \mathcal{H} -AMLS the Rayleigh Quotients $\widehat{\lambda}_j^{rq}$ have to be computed. However, the computational costs of this task are negligible compared to the other tasks because only $2n_{ev}$ sparse matrix-vector multiplications have to be applied.

6.2 Accuracy of the Eigenpair Approximation

The downside of faster computations and reduced memory requirements in \mathcal{H} -AMLS — achieved by a coarsening of the \mathcal{H} -matrix accuracy ε and a reduction of the truncation bound ω — is a possible loss in quality of the eigenpair approximations.

Keeping in mind the initial problem, the Rayleigh quotients $\widehat{\lambda}_j^{rq}$ in (26) are used to approximate the n_{ev} smallest eigenvalues λ_j of the continuous problem (2). For the approximation error it holds

$$\underbrace{|\lambda_j - \widehat{\lambda}_j^{rq}|}_{\text{error of the } \mathcal{H}\text{-AMLS method}} \leq \underbrace{|\lambda_j - \lambda_j^{(N)}|}_{\text{error caused by the discretisation}} + \underbrace{|\lambda_j^{(N)} - \widehat{\lambda}_j|}_{\text{error caused by the modal truncation}} + \underbrace{|\widehat{\lambda}_j - \widehat{\lambda}_j^{rq}|}_{\text{error caused by the } \mathcal{H}\text{-matrix approximation}}$$

where $\lambda_j^{(N)}$ is the eigenvalue of the discrete problem (4) and $\widehat{\lambda}_j$ is the eigenvalue of the reduced problem (18) from classical AMLS. The upper index of $\lambda_j^{(N)}$ is indicating the DOF of the underlying finite element space V_N . The approximation error of the \mathcal{H} -AMLS method is associated with the finite element discretisation, the modal truncation, and the \mathcal{H} -matrix approximation. The error caused by the modal truncation is influenced by the selected

eigenpairs of the partial eigensolution in (24) or respectively the truncation bound ω (cf. Remark 4), and the error caused by the use of \mathcal{H} -matrix approximation is influenced by the chosen accuracy ε .

In contrast to the \mathcal{H} -AMLS method, the approximation error of classical approaches, like the SIL algorithm, is only associated with the finite element discretisation because exact eigenvalues $\lambda_j^{(N)}$ of the discrete problem (4) are computed. The corresponding discretisation errors become reference values for the \mathcal{H} -AMLS method. To compete with a classical approach, the error caused by the modal truncation and the error caused by the use of the \mathcal{H} -matrix approximation have to be small enough that the error of the \mathcal{H} -AMLS method is of the same order as the discretisation error

$$\underbrace{|\lambda_j - \widehat{\lambda}_j^{rq}|}_{\text{error of } \mathcal{H}\text{-AMLS}} \approx \underbrace{|\lambda_j - \lambda_j^{(N)}|}_{\text{discretisation error}}. \quad (27)$$

Dividing (27) by $|\lambda_j|$ we obtain the equivalent statement expressed in form of relative errors

$$\widehat{\delta}_j^{rq} := \underbrace{\frac{|\lambda_j - \widehat{\lambda}_j^{rq}|}{|\lambda_j|}}_{\substack{\text{relative error} \\ \text{of } \mathcal{H}\text{-AMLS}}} \approx \underbrace{\frac{|\lambda_j - \lambda_j^{(N)}|}{|\lambda_j|}}_{\substack{\text{relative error} \\ \text{of discretisation}}} =: \delta_j^{(N)}. \quad (28)$$

In the following the aim is to choose the parameters ω and ε in such a way that (28) holds while the computational costs and storage requirements of \mathcal{H} -AMLS are reduced as much as possible.

7 Numerical Results

We analyse numerically the \mathcal{H} -AMLS method in the following for the Laplace eigenvalue problem

$$\begin{cases} -\Delta u = \lambda u & \text{in } \Omega = (0, 1)^3, \\ u = 0 & \text{on } \partial\Omega. \end{cases} \quad (29)$$

Note that the domain Ω is three-dimensional and it is very costly to solve this problem by the classical AMLS method (cf. Section 4). The eigenvalues in (29) are

$$\lambda = \lambda^{(\alpha, \beta, \gamma)} := \pi^2(\alpha^2 + \beta^2 + \gamma^2) \quad \text{with } \alpha, \beta, \gamma \in \mathbb{N}$$

and correspondingly it is possible to evaluate the relative errors $\delta_j^{(N)}$ and $\widehat{\delta}_j^{rq}$ from (28).

To solve problem (29) by a classical approach or \mathcal{H} -AMLS it is discretised first using standard P1 finite elements as described in Section 4. A discrete eigenvalue problem of the form (4) is obtained of the size $N = n^3$. The size N represents the DOF and $h := 1/(n+1)$ the *mesh width* of the model. The eigenvalues $\lambda_j^{(N)}$ of the discrete problem (4) are approximating the sought smallest n_{ev} eigenvalues λ_j of the continuous problem (29). The approximation errors of $\lambda_j^{(N)}$ depend on h . This issue is illustrated in Table 2 and Figure 8 where the mesh widths are

j	λ_j	error $ \lambda_j - \lambda_j^{(N)} $			rel. error $\delta_j^{(N)}$			$\max\{\delta_i^{(N)} \mid i = 1, \dots, j\}$		
		h_1	h_2	h_3	h_1	h_2	h_3	h_1	h_2	h_3
1	29.60	0.30	0.07	0.01	1.02e-2	2.57e-3	6.42e-4	1.02e-2	2.57e-3	6.42e-4
2	59.21	0.92	0.23	0.05	1.55e-2	3.88e-3	9.71e-4	1.55e-2	3.88e-3	9.71e-4
3	59.21	0.92	0.23	0.05	1.55e-2	3.88e-3	9.71e-4	1.55e-2	3.88e-3	9.71e-4
4	59.21	1.45	0.36	0.09	2.45e-2	6.11e-3	1.52e-3	2.45e-2	6.11e-3	1.52e-3
5	88.82	2.34	0.58	0.14	2.64e-2	6.62e-3	1.65e-3	2.64e-2	6.62e-3	1.65e-3
10	108.56	3.31	0.81	0.20	3.05e-2	7.48e-3	1.86e-3	3.50e-2	8.83e-3	2.21e-3
50	286.21	21.27	5.46	1.37	7.43e-2	1.91e-2	4.81e-3	1.01e-1	2.51e-2	6.27e-3
100	414.52	63.52	16.69	4.19	1.53e-1	4.02e-2	1.01e-2	1.53e-1	4.06e-2	1.09e-2
300	819.17	188.28	37.64	9.20	2.29e-1	4.59e-2	1.12e-2	2.60e-1	7.29e-2	2.20e-2

Table 2: Errors between the eigenvalues λ_j of the continuous problem (29) and the eigenvalues $\lambda_j^{(N)}$ of the discretised problem (K, M) for varying mesh widths. (All values given in this and the following tables are correct to two digits.)

- $h_1 := 0.05 \Rightarrow N = 6,859$,
- $h_2 := 0.025 \Rightarrow N = 59,319$,
- $h_3 := 0.0125 \Rightarrow N = 493,039$.

Halving the mesh width reduces the errors by a factor of approximately 4 but at the same time the system size increases by a factor of 8. Furthermore, it can be seen that smaller eigenvalues are better approximated than larger ones as already mentioned in Section 2. The relative errors $\delta_j^{(N)}$ resemble to an increasing sequence in j . Additionally it can be seen (cf. the last three columns of Table 2) that a finer mesh width is necessary to approximate more eigenvalues with the same accuracy. For example, to compute the smallest 10 eigenvalues with a relative accuracy of 1e-2 the mesh width h_2 is sufficient while for the smallest 300 eigenvalues a mesh width finer than h_3 is necessary and correspondingly more than 493,039 DOF are needed.

As a reference the eigenvalues $\lambda_j^{(N)}$ of the discrete problem have been computed by a shift-invert version of the subspace iteration¹ where the arising shift-invert systems have been solved with the help of the fast \mathcal{H} -matrix algebra. As discussed in Section 5 the \mathcal{H} -matrix format can be applied to the mass and stiffness matrix, and preconditioners for the corresponding shift-invert systems can be computed in almost linear complexity $\mathcal{O}(N \log^\alpha N)$ (cf. [13, 7]). We prefer the \mathcal{H} -matrix approach over sparse direct solvers because they can be expensive when the spatial domain is three-dimensional. The \mathcal{H} -matrix version of the shift-invert subspace iteration is called \mathcal{H} -SIS.

The computational costs of \mathcal{H} -SIS for computing the n_{ev} smallest eigenpairs are mainly composed by the costs for the vector iteration (done in $\mathcal{O}(C_{it} n_{ev} N \log N)$ using \mathcal{H} -matrix algebra), by the costs for computing the preconditioners (done in $\mathcal{O}(n_{ev} N \log^\alpha N)$ using \mathcal{H} -matrix algebra), and by the computational costs for maintaining the orthogonality of the iteration vectors (done in $\mathcal{O}(C_{it} n_{ev}^2 N)$ where $C_{it} > 0$ is the average number of matrix-vector multiplications needed for the convergence of one iteration-vector).

¹The subspace iteration is also called orthogonal iteration or simultaneous iteration.

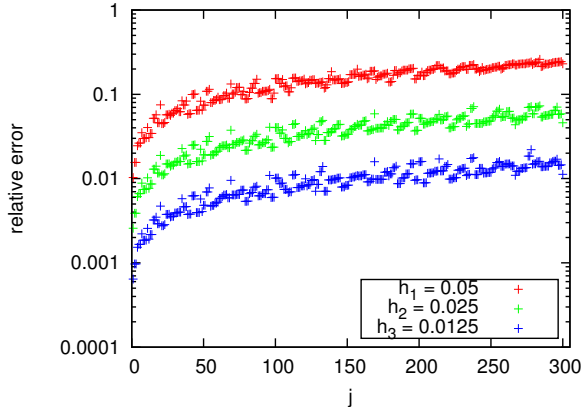


Figure 8: Relative errors $\delta_j^{(N)}$ of the smallest 300 eigenvalues of the problem (29) for varying mesh widths h .

There is a wide range of classical approaches and possible improvements for \mathcal{H} -SIS. For example a Block Lanczos process can be used instead of a subspace iteration as it is done in the SIL algorithm [16], or the computational costs for the orthogonalisation can be reduced by applying the reorthogonalisation as infrequently as possible and only for those vectors for which it is necessary (cf. [1]).

Neglecting possible computational costs for the orthogonalisation and the preconditioners, however, the best possible computational complexity of standard approaches would be

$$\mathcal{O}(C_{\text{it}} n_{ev} N). \quad (30)$$

These costs are at least necessary for the matrix-vector multiplication until the iteration-vectors converge to the smallest sought n_{ev} eigenvectors, i.e., the computational costs are increasing at least linearly in n_{ev} . We will show by numerical experiments that we achieve this complexity by \mathcal{H} -AMLS.

Applying \mathcal{H} -AMLS, the discrete problem (4) is projected onto a subspace using the fast \mathcal{H} -matrix algebra and the \mathcal{H} -reduced eigenvalue problem (25) is obtained where the Rayleigh quotients $\widehat{\lambda}_j^{rq}$ are approximating the sought eigenvalues λ_j of (29). Beside the DOF of the model, the relative errors $\widehat{\delta}_j^{rq}$ depend on the truncation bound ω and the chosen accuracy ε of the \mathcal{H} -matrix approximation. In the following we investigate how these parameters have to be chosen so that the eigenvalue approximations of \mathcal{H} -AMLS compete with those computed by \mathcal{H} -SIS. In particular we will test for $n_{ev} = 10, 50, 100, 300$ how the parameters have to be selected so that the inequality

$$\gamma_{n_{ev}}^{(N)} < 3 \quad (31)$$

holds where

$$\gamma_{n_{ev}}^{(N)} := \max \left\{ \widehat{\delta}_j^{rq} / \widehat{\delta}_j^{(N)} \mid j = 1, \dots, n_{ev} \right\}$$

is the maximal ratio between the relative error $\delta_j^{(N)}$ associated to \mathcal{H} -SIS (or another classical approach) and the relative error $\widehat{\delta}_j^{rq}$ associated to \mathcal{H} -AMLS. If inequality (31) is fulfilled it can be said that the approximation error of \mathcal{H} -AMLS is of the same order as the approximation error of a classical approach as claimed in (28).

7.1 Influence of the Modal Truncation

At first only the influence of the truncation bound ω is observed. To do this the \mathcal{H} -matrix approximation is deactivated in (23) by setting the parameter η from (21) to $\eta = 0$. Correspondingly no subblock is admissible, no Rk -matrix approximation is applied and the block diagonalisation of K and the matrix transformation of M in (23) are computed exactly (up to machine precision). In this situation \mathcal{H} -AMLS is equivalent with the classical AMLS method and correspondingly the computations will be very expensive as described in Section 4.

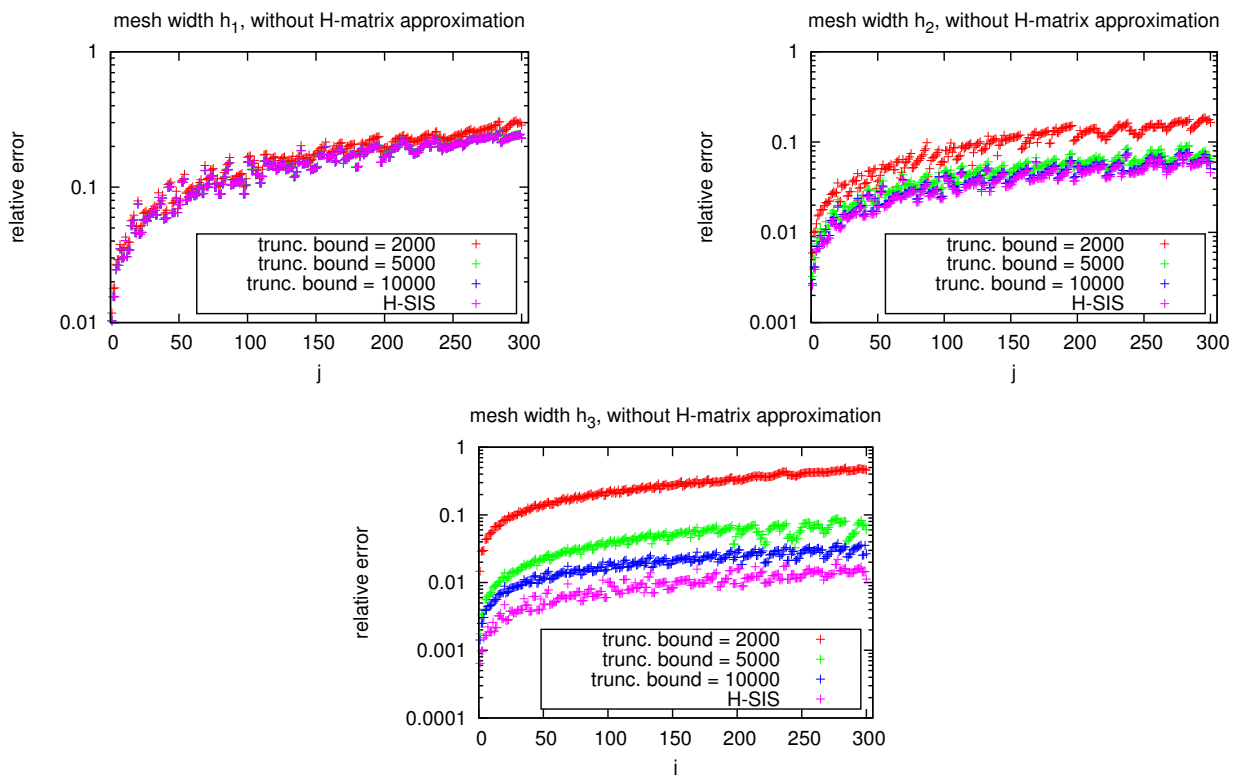


Figure 9: Influence of the truncation bound ω to the relative errors $\widehat{\delta}_j^{rq}$ of \mathcal{H} -AMLS for varying mesh widths and comparison with the relative errors $\delta_j^{(N)}$ of the classical approach \mathcal{H} -SIS. In this tests the \mathcal{H} -matrix approximation was deactivated (η was set to 0).

In Figure 9 the relative errors $\widehat{\delta}_j^{rq}$ and in Table 3 the maximal ratios $\gamma_{n_{ev}}^{(N)}$ are displayed for the mesh widths h_1, h_2 and h_3 using the truncation bounds

$$\omega_1 := 2000, \omega_2 := 5000 \quad \text{and} \quad \omega_3 := 10000.$$

For comparison with the classical approach the errors $\widehat{\delta}_j^{(N)}$ are displayed as well in Figure 9. In Table 3 can be seen that for h_1 the truncation bound ω_1 is sufficient in such a way that for all $n_{ev} = 10, 50, 100, 300$ the postulation (31) is fulfilled. For h_2 the bound ω_2 is sufficient and for h_3 the bound ω_3 . However, the computational costs are getting very expensive with increasing DOF because $\eta = 0$.

n_{ev}	$\gamma_{n_{ev}}^{(N)}$ for h_1			$\gamma_{n_{ev}}^{(N)}$ for h_2			$\gamma_{n_{ev}}^{(N)}$ for h_3		
	ω_1	ω_2	ω_3	ω_1	ω_2	ω_3	ω_1	ω_2	ω_3
10	1.17	1.00	1.00	2.66	1.34	1.06	30.03	3.85	2.58
50	1.22	1.01	1.00	3.08	1.42	1.08	34.64	5.45	2.78
100	1.22	1.01	1.00	3.08	1.42	1.08	35.52	6.22	2.78
300	1.28	1.02	1.00	3.57	1.46	1.11	40.58	6.47	2.78

Table 3: Influence of the truncation bound ω to the maximal ratios $\gamma_{n_{ev}}^{(N)}$ for varying mesh widths. In this tests the \mathcal{H} -matrix approximation was deactivated (η was set to 0).

n_{ev}	$\gamma_{n_{ev}}^{(N)}$ for h_1 using ω_1			$\gamma_{n_{ev}}^{(N)}$ for h_2 using ω_2			$\gamma_{n_{ev}}^{(N)}$ for h_3 using ω_3		
	ε_1	ε_2	$\eta = 0$	ε_1	ε_2	$\eta = 0$	ε_1	ε_2	$\eta = 0$
10	1.21	1.17	1.17	3.63	1.35	1.34	44.27	2.69	2.58
50	1.22	1.22	1.22	3.63	1.43	1.42	44.27	2.87	2.78
100	1.22	1.22	1.22	3.63	1.43	1.42	44.27	2.87	2.78
300	1.29	1.28	1.28	3.63	1.46	1.46	44.27	2.87	2.78

Table 4: Influence of the \mathcal{H} -matrix approximation accuracy ε to the maximal ratios $\gamma_{n_{ev}}^{(N)}$ for varying mesh widths.

7.2 Influence of the \mathcal{H} -Matrix Approximation

To speed up the computations of the block diagonalisation of K and the matrix transformation of M in (23) the \mathcal{H} -matrix approximation is activated by setting the parameter η in (21) back to $\eta = 50$. Accordingly certain subblocks get admissible and corresponding submatrices are approximated by Rk -matrices with a given approximation accuracy ε .

In the passage before could be seen that the truncation bound ω_1 was sufficient for the mesh width h_1 , ω_2 for the width h_2 and ω_3 for the width h_3 . Using this different truncation bounds the computations have been done again applying the \mathcal{H} -matrix approximation accuracies

$$\varepsilon_1 := 1e-1 \quad \text{and} \quad \varepsilon_2 := 1e-2.$$

The corresponding relative errors $\widehat{\delta}_j^{rq}$ are displayed in Figure 10 and in Table 4 the maximal ratios $\gamma_{n_{ev}}^{(N)}$. In Table 4 can be seen that for h_1 the truncation bound ω_1 and the accuracy ε_1 are sufficient to fulfil postulation (31) for all $n_{ev} = 10, 50, 100, 300$. For h_2 are the parameters ω_2 and ε_2 sufficient, and for h_3 the parameters ω_3 and ε_2 .

In Figure 9 and Figure 10 can be observed that for smaller mesh widths larger truncation bounds ω and better \mathcal{H} -matrix approximation accuracies ε are necessary to fulfil postulation (31). Furthermore, in Figure 10, with the mesh width h_3 , can be seen that the approximation of smaller eigenvalues behaves more sensitive to the chosen accuracy ε than the approximation of larger eigenvalues.

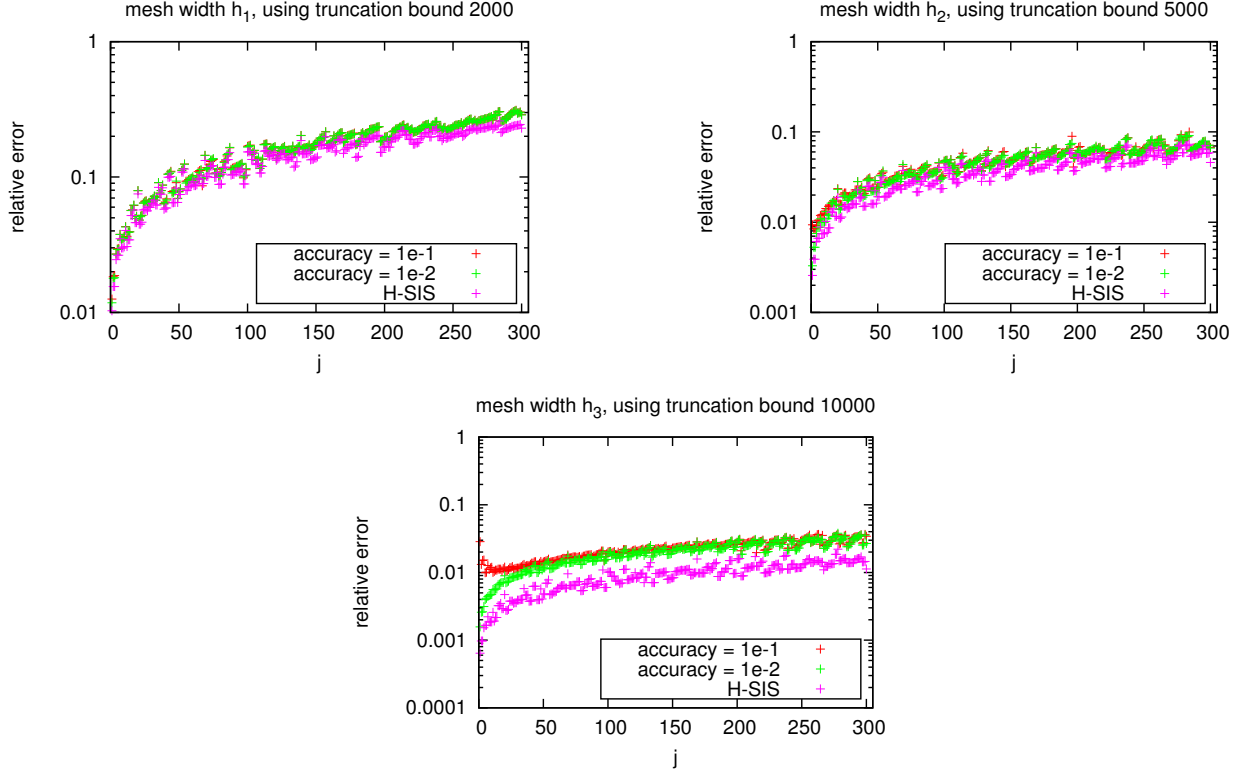


Figure 10: Influence of the \mathcal{H} -matrix approximation accuracy ε to the relative errors $\widehat{\delta}_j^{rq}$ of \mathcal{H} -AMLS for varying mesh widths and comparison with the relative errors $\delta_j^{(N)}$ of the classical approach \mathcal{H} -SIS.

7.3 Analysis of the Computational Time

The computational costs of the benchmarks from the previous section, using sufficient parameters ω and ε , are given in Table 5 for $n_{ev} = 300$. The costs of the different tasks (indicated in Table 1) are displayed there, and the order k of the \mathcal{H} -reduced eigenvalue problem $(\widehat{\mathbf{K}}, \widehat{\mathbf{M}})$. The computation of the block diagonalisation $K = L^{\mathcal{H}} \widetilde{K}^{\mathcal{H}} (L^{\mathcal{H}})^T$ and the matrix transformation $\widetilde{M}^{\mathcal{H}} = (L^{\mathcal{H}})^{-1} M (L^{\mathcal{H}})^{-T}$, task **(T2)** and **(T3)**, are dominating the costs of the other tasks. In these benchmarks the eigenpairs of the \mathcal{H} -reduced problem have been computed by the dense eigensolver `dsygvx` of LAPACK. However, if the size k of the \mathcal{H} -reduced problem is getting larger a sparse eigensolver, e.g. provided by ARPACK, should be used because the matrices $\widehat{\mathbf{K}}$ and $\widehat{\mathbf{M}}$ are sparse and only the smallest n_{ev} eigenpairs have to be computed (cf. Remark 5 ii)). Correspondingly in Table 5 can be observed that the computational costs of task **(T7)** are increasing stronger than the others.

The parameters ω and ε in the benchmarks have been chosen in such a way that they fulfil postulation (31), i.e., that the eigenvalue approximations computed by \mathcal{H} -AMLS are of the same quality as approximations computed by a classical approach. In Table 4 can be seen that the ratios between $\widehat{\delta}_j^{rq}$ and $\widehat{\delta}_j^{(N)}$ are only slowly increasing in j . It looks like that ω and ε have to be chosen rather in dependency to the mesh width h than to the number

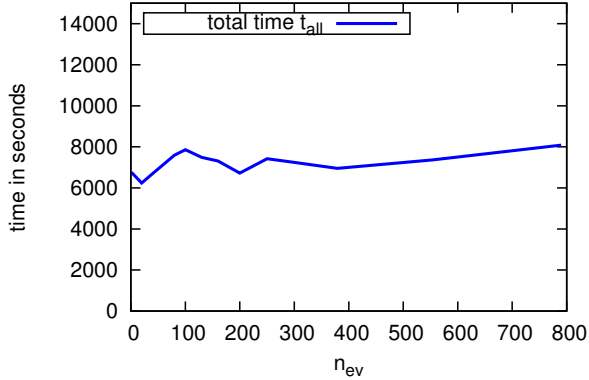


Figure 11: Computational time of \mathcal{H} -AMLS for computing the smallest $n_{ev} = 1, \dots, 790 = 10N^{1/3}$ eigenpairs for the mesh width h_3 using the parameters ω_3 and ε_2 .

	parameter		problem size		time for computing (percentage of total time)			time	
	ω	ε	N	k	task (T2)–(T3)	task (T4)–(T6)	task (T7)–(T8)	t_{all}	avg(t_{all})
h_1	2,000	1e-1	6,859	776	9.78s (67.4%)	2.23s (15.3%)	1.90s (13.1%)	14.50s	7.04s
h_2	5,000	1e-2	59,319	3,909	192 (50.2%)	47s (12.3%)	136s (35.6%)	382s	21.45s
h_3	10,000	1e-2	493,039	11,679	2,636s (43.9%)	1,498s (24.9%)	1,816s (30.2%)	5,998s	40.55s

Table 5: Computational costs of \mathcal{H} -AMLS computing the smallest $n_{ev} = 300$ eigenpairs for varying mesh widths using sufficient parameters to fulfil postulation (31). t_{all} is the total time needed for the computations and $avg(t_{all})$ the average time defined in (32) using $n_{ev} = 300$.

of sought eigenvalues n_{ev} to fulfil (31). It seems that in these benchmarks much more than 300 eigenvalue approximations can be computed with nearly the same approximation quality as of an classical approach. Increasing the number of sought eigenpairs, however, increases just slightly the computational costs of \mathcal{H} -AMLS as it can be seen in Figure 11. This is a big advantage to classical approaches whose computational costs are at least of the order $\mathcal{O}(C_{it} n_{ev} N)$, i.e., whose costs at least linearly increasing in n_{ev} (cf. (30)). This is very useful particularly because we are interested in computing a large amount of eigenpairs, e.g. the first $n_{ev} = CN^{1/3}$ or $n_{ev} = CN^{1/2}$ ones for some $C > 0$, as discussed in Section 2.

In this context a possible measure for the performance of an eigensolver is the needed computational time per eigenpair and per one Million DOF, formally defined by $avg(t_{all})$, where t_{all} is the total time needed for the computation of the first n_{ev} eigenpairs and

$$avg(t) := avg(t, n_{ev}, N) := \frac{10^6 t}{N n_{ev}}. \quad (32)$$

Assume for example that a classical approach has the best possible complexity $\mathcal{O}(C_{it} n_{ev} N)$, where in average 10 iterations are necessary until an iteration vector converges, and the matrix-vector multiplication takes 5 seconds per one million DOF (experience value from practice is 1s–10s, depending on the computer and without parallelisation) then the average computational time of this eigensolver is $avg(t_{all}) = 50s$.

To get an impression of the average computational time of \mathcal{H} -AMLS, we want to determine $avg(t_{all})$ for varying N where the number of sought eigenpairs is set to $n_{ev} := 10N^{1/3}$. The \mathcal{H} -matrix accuracy ε and the truncation bound ω , respectively, the number of selected eigenpairs

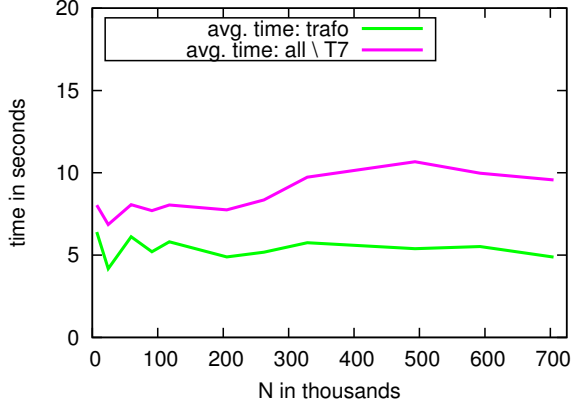


Figure 12: Average computational times $\text{avg}(t_{\text{all}\setminus\text{T7}})$ and $\text{avg}(t_{\text{trafo}})$ using $n_{ev} = 10N^{1/3}$.

k_i in (24) have to be chosen according to the DOF, as discussed in the previous sections, to obtain eigenpair approximation of comparable quality to classical approaches. As orientation we use the parameter settings from the benchmarks before. We set $\varepsilon = 1e-1$ for the mesh width h_1 and divide ε by three when h is halved. In stead of adjusting ω , we set the number of selected eigenpairs to $k_i := (N_i)^{1/3}$ if the subproblem is associated to a subdomain and to $k_i := (N_i)^{1/2}$ if it is associated to an interface. Using this parameter setting postulation (31) was fulfilled for the mesh widths h_1 , h_2 and h_3 .

Because in our implementation the dense eigensolver `dsygvx` is used instead of a sparse solver as recommended the computational costs of task **(T7)** are omitted. Correspondingly we investigate the average computational time $\text{avg}(t_{\text{all}\setminus\text{T7}})$ using definition (32) with $n_{ev} = 10N^{1/3}$, where $t_{\text{all}\setminus\text{T7}}$ is the total computational time minus the time needed for task **(T7)**. In Figure 12 can be seen that this time is nearly constant which shows — without considering the time for task **(T7)** — that \mathcal{H} -AMLS computes the first $n_{ev} = N^{1/3}$ eigenpairs with an optimal complexity of $\mathcal{O}(Nn_{ev})$.

Furthermore we investigated in Figure 12 the average computational time $\text{avg}(t_{\text{trafo}})$ where t_{trafo} is the time needed for the block diagonalisation $K = L^{\mathcal{H}}\tilde{K}^{\mathcal{H}}(L^{\mathcal{H}})^T$ and the matrix transformation $\tilde{M}^{\mathcal{H}} = (L^{\mathcal{H}})^{-1}M(L^{\mathcal{H}})^{-T}$, task **(T2)** and **(T3)**. These both tasks seem to be the computational bottleneck of the \mathcal{H} -AMLS method. It can be observed that the average time $\text{avg}(t_{\text{trafo}})$ is slightly decreasing, and it even tends to zero because the computational costs of these tasks are of the order $\mathcal{O}(N \log^{\alpha} N)$.

8 Conclusion

To solve an elliptic PDE eigenvalue problem we have combined the automated multi-level substructuring with the concept of hierarchical matrices. Whereas the classical AMLS method is very effective in the two-dimensional case, it is getting very expensive for three-dimensional problems. The required computation of the transformed eigenvalue problem (\tilde{K}, \tilde{M}) is one computational bottleneck of the classical AMLS method in the three-dimensional case. Using the fast \mathcal{H} -matrix algebra, however, we can compute the transformed eigenvalue problem very efficiently in almost linear complexity $\mathcal{O}(N \log^{\alpha} N)$ which is even independent of the number of sought eigenpairs. Also the computation of the partial eigensolutions $(\tilde{K}_{ii}, \tilde{M}_{ii})$

and the reduced eigenvalue problem $(\widehat{K}, \widehat{M})$ are performed much more efficiently using the fast \mathcal{H} -matrix algebra. Altogether the new \mathcal{H} -AMLS method allows us to compute a large amount of eigenpair approximations in almost optimal complexity.

\mathcal{H} -AMLS has to be benchmarked in further examples, especially for problems arising from applications. However, the numerical results demonstrate the potential of the method in solving large-scale elliptic PDE eigenvalue problems. Moreover, \mathcal{H} -AMLS is well suited for parallelisation. Last but not least, a recursively applied version of \mathcal{H} -AMLS might further reduce the overall computational costs and lead to a solver of complexity $o(Nn_{ev})$.

References

- [1] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst, editors. *Templates for the solution of algebraic eigenvalue problems*, volume 11 of *Software, Environments, and Tools*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000. A practical guide.
- [2] L. Banjai, S. Börm, and S. Sauter. FEM for elliptic eigenvalue problems: how coarse can the coarsest mesh be chosen? An experimental study. *Comput. Vis. Sci.*, 11(4-6):363–372, 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] J. K. Bennighof. Adaptive multi-level substructuring method for acoustic radiation and scattering from complex structures. *Computational Methods for Fluid/Structure Interaction*, 178:25–38, 1993.
- [5] J. K. Bennighof, M. F. Kaplan, and M. B. Muller. Extending the frequency response capabilities of automated multi-level substructuring. *AIAA Dynamics Specialists Conference*, 2000. AIAA Paper 2000-1574.
- [6] J. K. Bennighof and R. B. Lehoucq. An automated multilevel substructuring method for eigenspace computation in linear elastodynamics. *SIAM J. Sci. Comput.*, 25(6):2084–2106 (electronic), 2004.
- [7] S. L. Borne and L. Grasedyck. H-matrix preconditioners in convection-dominated problems. *SIAM J. Matrix Analysis Applications*, 27(4):1172–1183, 2006.
- [8] F. Bourquin. Analysis and comparison of several component mode synthesis methods on one-dimensional domains. *Numerische Mathematik*, 58:11–34, 1990.
- [9] F. Bourquin. Component mode synthesis and eigenvalues of second order operators: Discretization and algorithm. *Mathematical Modeling and Numerical Analysis*, 26:385–423, 1992.

- [10] R. R. Craig and M. C. C. Bampton. Coupling of substructures for dynamic analysis. *AIAA Journal*, 6:1313–1319, 1968.
- [11] K. Elssel and H. Voss. An a priori bound for automated multi-level substructuring. Technical report 81, TU Hamburg-Harburg, 2004.
- [12] W. Gao, X. S. Li, C. Yang, and Z. Bai. An implementation and evaluation of the AMLS method for sparse eigenvalue problems. *ACM Trans. Math. Software*, 34(4):Art. 20, 28, 2008.
- [13] L. Grasedyck and W. Hackbusch. Construction and arithmetics of \mathcal{H} -matrices. *Computing*, 70(4):295–334, 2003.
- [14] L. Grasedyck, R. Kriemann, and S. Le Borne. Parallel black box \mathcal{H} -LU preconditioning for elliptic boundary value problems. *Comput. Vis. Sci.*, 11(4-6):273–291, 2008.
- [15] L. Grasedyck, R. Kriemann, and S. LeBorne. Domain decomposition based \mathcal{H} -LU preconditioning. *Numerische Mathematik*, 112(4):565–600, 2009.
- [16] R. G. Grimes, J. G. Lewis, and H. D. Simon. A shifted block Lanczos algorithm for solving sparse symmetric generalized eigenproblems. *SIAM J. Matrix Anal. Appl.*, 15(1):228–272, 1994.
- [17] W. Hackbusch. *Elliptic differential equations : theory and numerical treatment*, volume 18 of *Springer series in computational mathematics*. Springer, Berlin, 1992.
- [18] W. Hackbusch. A sparse matrix arithmetic based on \mathcal{H} -matrices. Part I: Introduction to \mathcal{H} -matrices. *Computing*, 62(2):89–108, 1999.
- [19] W. Hackbusch. *Hierarchische Matrizen : Algorithmen und Analysis*. Springer, Dordrecht, 2009.
- [20] U. Hetmaniuk and R. B. Lehoucq. Multilevel methods for eigenspace computations in structural dynamics. In *In Domain Decomposition Methods in Science and Engineering*, pages 103–114. Springer-Verlag, 2007.
- [21] W. C. Hurty. Vibrations of structural systems by component-mode synthesis. *Journal of the Engineering Mechanics Division*, 86:51–69, 1960.
- [22] M. F. Kaplan. *Implementation of automated multi-level substructuring for frequency response analysis of structures*. Ph.d. thesis, Universtiy of Texas at Austin, 2001.
- [23] A. Kropp and D. Heiserer. Efficient broadband vibro-acoustic analysis of passenger car bodies using an fe-based component mode synthesis approach. *Journal of Computational Acoustics*, 11(02):139–157, 2003.
- [24] A. Quarteroni and A. Valli. *Domain decomposition methods for partial differential equations*. Numerical Mathematics and Scientific Computation. The Clarendon Press Oxford University Press, New York, 1999. Oxford Science Publications.

- [25] S. Sauter. hp-finite elements for elliptic eigenvalue problems: Error estimates which are explicit with respect to λ , h , and p . *SIAM J. Numerical Analysis*, 48(1):95–108, 2010.
- [26] P. Seshu. Substructuring and component mode synthesis. *Shock and Vibration*, 4:199–210, 1997.
- [27] C. Yang, W. Gao, Z. Bai, X. S. Li, L.-Q. Lee, P. Husbands, and E. Ng. An algebraic substructuring method for large-scale eigenvalue calculation. *SIAM J. Sci. Comput.*, 27(3):873–892 (electronic), 2005.