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

(October 2014, accepted for publication in *Computing and Visualization in Science*)

Peter Gerds[*], Lars Grasedyck[*]

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 optimal 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 [4, 6, 25] 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 [24] and further improved by Craig and Bampton [11]. During the years CMS became very popular and was studied by many researchers, e.g. in [8, 9, 10] a mathematical analysis of CMS is given and in [32] an overview over different versions of CMS.

The single-level substructuring performed in CMS is extended in AMLS to a multi-level version. 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 of much smaller size 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, 25, 28] 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. [23]). 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 [18] which is commonly used in structural engineering. Kropp and Heiserer presented breakthrough calculations in [28]. They benchmarked the AMLS method 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 introduce a new approach called $\mathcal{H}$-AMLS which is a combination of the AMLS method and $\mathcal{H}$-matrices. $\mathcal{H}$-matrices [20, 21] are a data-sparse approximation of dense matrices which e.g. result from the inversion [3, 13] or the LU-factorisation [2, 13, 16, 29] 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 [15, 17]. In the new method this fast $\mathcal{H}$-matrix algebra is used to setup the reduced eigenvalue problem and thus $\mathcal{H}$-AMLS is well-suited for three-dimensional problems. In the three dimensional case it is essential to keep the size of the reduced problem small. We achieve this by a new recursive formulation of AMLS. Eventually it turns out that all previously expensive steps of AMLS can be performed in almost linear complexity (linear up to logarithmic factors) in the size N of the discrete problem. The remaining bottleneck is more of a theoretical nature: the setup of the reduced problem and the extraction of the eigenvectors from the solutions of the reduced problem require n_{ev}^2 scalar products of length N, where n_{ev} is the number of sought eigenvectors. Due to the very small constant involved in these computations, their effect on practical computations is hardly visible.

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

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

where Ω is a d-dimensional domain $(d = 2, 3)$ with a Lipschitz boundary $\partial\Omega$ and L is a uniformly elliptic second order partial differential operator in divergence form

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

with $L^\infty(\Omega)$-functions a_{ij}, i.e. for $x \in \Omega$ the matrix $A(x) := \big(a_{ij}(x)\big)_{i,j=1}^{d}$ is symmetric positive definite and the eigenvalues of $A(x)$ are uniformly bounded from below by a positive constant. In weak formulation (1) can be expressed as

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

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

According to [19] the continuous eigenvalue problem (2) possesses a countable family of eigensolutions

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

with eigenvalues λ_j ordered such that $\lambda_j \leq \lambda_{j+1}$.

We approximate solutions of the continuous eigenvalue problem by discretisation. Using an N-dimensional finite element space denoted by $V_N \subset H^1_0(\Omega)$ and spanned by its basis functions $\big(\varphi_i^{(N)}\big)_{i=1}^N$ the continuous eigenvalue problem (2) is discretised by

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

where the stiffness matrix

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

and the mass matrix

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

are both sparse, symmetric and positive definite. The eigenvalues of (4) are positive and the corresponding eigenpairs $\big(\lambda_j^{(N)}, x_j^{(N)}\big)_{j=1}^N \in \mathbb{R}_{>0} \times \mathbb{R}^N$ 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. [19], it holds $\lambda_j^{(N)} \to \lambda_j$ for $N \to \infty$ and — assuming that $(u_j^{(N)}, u_j^{(N)})_0 = 1$ — a subsequence of $\big(u_j^{(N)}\big)_{N=1}^\infty$ is converging in $H^1_0(\Omega)$ to an eigenfunction $u \in E(\lambda_j)$ where $u_j^{(N)} \in V_N$ is defined by $u_j^{(N)} := \sum_{i=1}^N \big(x_j^{(N)}\big)_i \cdot \varphi_i^{(N)}$ and $E(\lambda_j) \subset H^1_0(\Omega)$ is the eigenspace of the continuous eigenvalue λ_j which is defined by

$$E(\lambda_j) := \operatorname{span}\Big\{ u \in H^1_0(\Omega) \;\Big|\; a(u,v) = \lambda_j (u,v)_0 \quad \forall\, v \in H^1_0(\Omega)\Big\}.$$

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. [1, 31], 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. Also, if the number of sought discrete eigenvalues approaches N, it is advisable to use either a cubic scaling direct method or an iterative method like SIL with a good shift strategy and an efficient solver for the arising shifted linear systems.

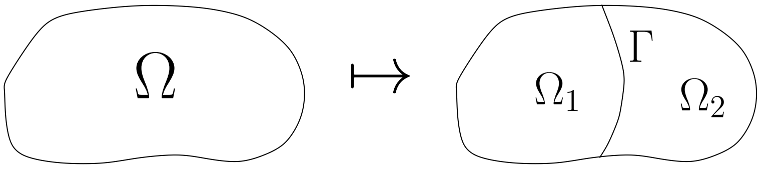


Figure 1: Partitioning of the domain Ω into two non-overlapping subdomains.

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 Version: Continuous Setting

The single-level version of 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 $\Gamma := \overline{\Omega}_1 \cap \overline{\Omega}_2$. 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

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

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

$$V_\Gamma := \Big\{ E_\Omega \tau \;\Big|\; \tau \in H_{00}^{1/2}(\Gamma) \Big\}.$$

Here $H_{00}^{1/2}(\Gamma)$ 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 $\tau \in H_{00}^{1/2}(\Gamma)$ which is defined as the unique solution of the Dirichlet problem

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

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 $H^1_0(\Omega)$.

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

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*

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

and for V_Γ the so-called *coupling mode eigenvalue problem*

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

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 $H^1_0(\Omega)$. Each of these problems possess a countable family of eigensolutions which are given by

$$\big(\lambda^{\Omega_i}_j \, , \, u^{\Omega_i}_j\big)^{\infty}_{j=1} \in \mathbb{R}_{>0} \times V_{\Omega_i} \quad \text{with } \lambda^{\Omega_i}_j \leq \lambda^{\Omega_i}_{j+1}$$

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

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

for the coupling mode eigenvalue problem (8). The eigenfunctions (7) and (8) form a basis of V_{Ω_i} and V_Γ. According to Theorem 1 these functions are a-orthogonal to each other and form a basis of $H^1_0(\Omega)$ with

$$H^1_0(\Omega) \ = \ \bigcup_{i=1}^{2} \operatorname{span}\Big\{ u^{\Omega_i}_j \ \big| \ j \in \mathbb{N} \Big\} \ \cup \ \operatorname{span}\Big\{ u^{\Gamma}_j \ \big| \ j \in \mathbb{N} \Big\}. \tag{9}$$

We remark that even if the eigensolutions of the problems (7) and (8) are known the global eigenvalue problem (2) is not solved. However, the eigenfunctions of (7) and (8) belonging to the smallest eigenvalues are well suited to approximate the sought eigensolutions $(\lambda_j, u_j)^{n_{ev}}_{j=1}$ of (2). This issue is reasoned by various numerical studies (see, e.g., [6]) and is motivated by the error analysis done in [9, 10] for a quite similar method. Correspondingly, to approximate the sought eigensolutions, in the third step of AMLS the finite dimensional subspace $U_k \subset H^1_0(\Omega)$ is defined by

$$U_k \ := \ \bigcup_{i=1}^{2} \operatorname{span}\Big\{ u^{\Omega_i}_j \ \big| \ j = 1, \ldots, k_i \Big\} \ \cup \ \operatorname{span}\Big\{ u^{\Gamma}_j \ \big| \ j = 1, \ldots, k_\Gamma \Big\} \tag{10}$$

which is obtained by applying a modal truncation in (9) and selecting only those eigenfunctions which belong to the smallest k_1, k_2 and k_Γ eigenvalues for given $k_1, k_2, k_\Gamma \in \mathbb{N}$ and $k = k_1 + k_2 + k_\Gamma$.

Furthermore, we note 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 $p(N) = N^{1/2}$, which motivates the modal truncation performed in (10) from another point of view.

Using the finite dimensional subspace U_k the so-called *reduced eigenvalue problem*

$$\begin{cases} \text{find } \ (\lambda^{(k)}, u^{(k)}) \in \mathbb{R} \times U_k \ \text{ such that} \\ a(u^{(k)}, v) \ = \ \lambda^{(k)} (u^{(k)}, v)_0 \quad \forall\, v \in U_k \end{cases} \tag{11}$$

is defined with the eigensolutions

$$\big(\lambda_j^{(k)}, u_j^{(k)}\big)_{j=1}^{k} \in \mathbb{R}_{>0} \times U_k \quad \text{with } \lambda_j^{(k)} \leq \lambda_{j+1}^{(k)} \tag{12}$$

which is the Ritz-Galerkin approximation of the original global eigenvalue problem (2).

Correspondingly in the fourth and last step of the AMLS method the first n_{ev} eigensolutions (12) are computed (with $n_{ev} \leq k$) which are approximating the sought eigensolutions $(\lambda_j, u_j)_{j=1}^{n_{ev}}$ of (2).

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

$$\begin{cases} \text{find } (\lambda, u) \in \mathbb{R} \times H_{00}^{1/2}(\Gamma) \ \text{ such that} \\ \langle \mathcal{S} u, v \rangle = \lambda \langle \mathcal{M} u, v \rangle \quad \forall v \in H_{00}^{1/2}(\Gamma) \end{cases} \tag{13}$$

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

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

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

$$\begin{cases} \text{find } \ \mathcal{G}_i(f) \in V_{\Omega_i} \ \text{ such that} \\ a(\mathcal{G}_i(f), v) = (f, v)_0 \ \ \forall\, v \in V_{\Omega_i}, \end{cases} \tag{14}$$

i.e., $\mathcal{G}_i$ is the Green's function of problem (14).

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 $H^1(\Omega)$-elliptic bilinear forms. The operator $\mathcal{M}$ is derived according to [6] and the operator $\mathcal{S}$ according to [30].

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. [30]). $\mathcal{M}$ 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*

$$\begin{cases} \text{find } (\lambda, u) \in \mathbb{R} \times H_0^1(\Omega_i) \text{ such that} \\ a(u,v) = \lambda (u,v)_0 \qquad \forall v \in H_0^1(\Omega_i). \end{cases} \tag{15}$$

iii) *The benefit of the representation* (15) *and* (13) *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 that, in order to solve the global eigenvalue problem, the domain Ω is partitioned into two subdomains which are separated by an interface. On the subdomains and on the interface suitable eigenvalue problems are defined which, however, do not solve the global problem but whose eigenfunctions are well suited to approximate the sought eigensolutions of the global problem. In particular eigenfunctions belonging to the smallest eigenvalues are selected from each subproblem to form a suitable subspace which is used for a Ritz-Galerkin approximation of the global problem. Finally, we obtain from the resulting reduced eigenvalue problem approximations of the sought eigensolutions of the global problem.

3.2 Single-Level Version: Algebraic Setting

In this section we describe AMLS in the algebraic setting to show how the method is applied in practice. The initial point is the discretised eigenvalue problem (4). For reasons of convenience we leave out in this particular section the upper index of $\lambda^{(N)}$ and $x^{(N)}$ in (4) — indicating the number of DOF of the finite element discretisation — and the following eigenvalue problem

$$\begin{cases} \text{find } (\lambda, x) \in \mathbb{R} \times \mathbb{R}^N \text{ with} \\ K\,x \;=\; \lambda\, M\, x \end{cases} \tag{16}$$

is considered with the eigenpairs $\big(\lambda_j, x_j\big)_{j=1}^{N} \in \mathbb{R}_{>0} \times \mathbb{R}^N$ and $\lambda_j \leq \lambda_{j+1}$.

Because the matrices K and M in (16) 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, in the first step of AMLS the row and column indices are reordered in such a way that

$$K \;=\; \begin{array}{c|ccc} & \Omega_1 & \Omega_2 & \Gamma \\ \hline \Omega_1 & K_{11} & & K_{13} \\ \Omega_2 & & K_{22} & K_{23} \\ \Gamma & K_{31} & K_{32} & K_{33} \end{array} \qquad \text{and} \qquad M \;=\; \begin{array}{c|ccc} & \Omega_1 & \Omega_2 & \Gamma \\ \hline \Omega_1 & M_{11} & & M_{13} \\ \Omega_2 & & M_{22} & M_{23} \\ \Gamma & M_{31} & M_{32} & M_{33} \end{array} \tag{17}$$

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 (17) 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 $K = L\widetilde{K}L^T$ with

$$L := \begin{bmatrix} \mathrm{Id} & & \\ & \mathrm{Id} & \\ K_{31}K_{11}^{-1} & K_{32}K_{22}^{-1} & \mathrm{Id} \end{bmatrix} \qquad \text{and} \qquad \widetilde{K} = \operatorname{diag}\big[K_{11}, K_{22}, \widetilde{K}_{33}\big].$$

The submatrix $\widetilde{K}_{33}$ given by

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

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

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

where the submatrices of $\widetilde{M}$ are given by

$$\widetilde{M}_{3i} = M_{3i} - K_{3i}K_{ii}^{-1}M_{ii}, \quad \widetilde{M}_{i3} = \widetilde{M}_{3i}^T \quad \text{for } i = 1, 2$$

and

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

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 from (K, M) to $(\widetilde{K}, \widetilde{M})$ is not obvious. But it can be shown, cf. [6] and [30], that the eigenvalue problem $(\widetilde{K}_{33}, \widetilde{M}_{33})$ is the discrete equivalent of the continuous coupling mode eigenvalue problem (13), and the eigenvalue problems (K_{11}, M_{11}) and (K_{22}, M_{22}) are the discrete equivalents of the continuous fixed-interface problems (15). As in the continuous setting the global eigenvalue problem (K, M), respectively $(\widetilde{K}, \widetilde{M})$, is not solved just by computing the eigensolution of the subproblems (K_{11}, M_{11}), (K_{22}, M_{22}) and $(\widetilde{K}_{33}, \widetilde{M}_{33})$. However, the eigenvectors of these three subproblems are well suited to approximate the sought eigenvectors of (K, M) and $(\widetilde{K}, \widetilde{M})$. As in the continuous setting, cf. (10), only those subproblem eigenvectors are of interest which belong to the smallest eigenvalues.

Correspondingly in the next step of AMLS partial eigensolutions of the subproblems are computed, i.e., 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

$$K_{ii}\, S_i = M_{ii}\, S_i\, D_i \quad \text{for } i = 1, 2 \qquad \text{and} \qquad \widetilde{K}_{33}\, S_3 = \widetilde{M}_{33}\, S_3\, D_3 \tag{18}$$

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

Remark 4 (Mode Selection) *How many eigenvectors have to be selected in* (18) *from each subproblem is not easy to answer. On the one hand enough spectral information has to be kept to obtain sufficiently good eigenpair approximations from the reduced problem. Selecting all (discrete) eigenvectors from each subproblem would lead to exact eigenpairs of the discrete global eigenvalue problem* (K, M). *On the other hand* k_i *should be small to obtain in the further proceeding of AMLS a reduced problem of small size which can be easily solved.*

In the literature [12, 33] several heuristic approaches 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* (1). *One possible strategy for the eigenpair selection in* (18) *is as follows: Select in each subproblem only those eigenpairs whose eigenvalues are smaller than a given* truncation bound $\omega > 0$.

We pursue a different approach here. The three subproblems (K_{11}, M_{11}), (K_{22}, M_{22}) *and* $(\widetilde{K}_{33}, \widetilde{M}_{33})$ *correspond to finite element discretisations of the continuous problems* (15) *and* (13) *(cf. [6, 30]). Therefore and because of the remark regarding the approximation property of finite element spaces in Section 2, all eigenvectors in* (18) *are computed which still lead to reasonable approximations of the corresponding continuous eigenfunctions. Correspondingly only the eigenvectors belonging, e.g., to the smallest*

$$k_i = C N_i^{1/3} \in \mathbb{N} \qquad \text{or} \qquad k_i = C N_i^{1/2} \in \mathbb{N}$$

eigenvalues are computed, for a constant $C > 0$ *that will be specified in Section 7.*

In the next step the block diagonal matrix

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

with $k := k_1 + k_2 + k_3 \ll N$ is defined which is built of all selected subproblem eigenvectors. The k-dimensional subspace spanned by the columns of the matrix S respectively of the matrix $L^{-T} S$ is well suited to approximate the sought eigenvectors of $(\widetilde{K}, \widetilde{M})$ respectively (K, M). More precisely, the columns of $L^{-T} S$ are the discrete equivalent of the selected eigenfunctions contained in subspace (10) from the continuous setting (cf. [6, 30]).

In order to approximate the sought eigenpairs of (K, M) in the next step of AMLS 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

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

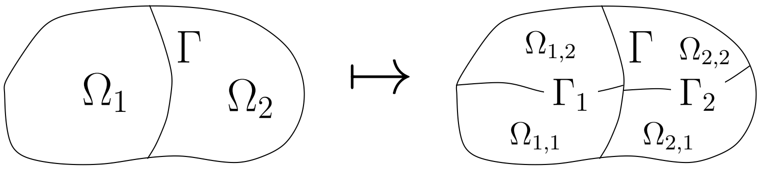


Figure 2: Extending the single-level substructuring of Ω to a two-level substructuring.

and a *reduced eigenvalue problem*

$$\begin{cases} \text{find } (\widehat{\lambda}, \widehat{x}\,) \in \mathbb{R} \times \mathbb{R}^k \text{ with} \\ \widehat{K}\,\widehat{x} \,=\, \widehat{\lambda}\,\widehat{M}\,\widehat{x} \end{cases} \tag{19}$$

with eigenpairs $\big(\widehat{\lambda}_j, \widehat{x}_j\big)_{j=1}^{k} \in \mathbb{R}_{>0} \times \mathbb{R}^k$ and $\widehat{\lambda}_j \leq \widehat{\lambda}_{j+1}$ is obtained. Here we note that the reduced eigenvalue problem (19) is the discrete equivalent of the reduced problem (11) from the continuous setting.

At the end of AMLS the smallest n_{ev} eigenpairs of (19) are computed. The vectors

$$\widehat{y}_j := L^{-T} S\, \widehat{x}_j \quad \text{ with } j = 1, \ldots, k \tag{20}$$

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

Remark 5 (Reduced Eigenvalue Problem) *Because the eigenpairs of the reduced eigenvalue problem $(\widehat{K}, \widehat{M})$ 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* (18)*. 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 $(\widehat{K}, \widehat{M})$ are of comparable quality as the eigenpair approximations derived from the problem (K, M).*

The reduced eigenvalue problem $(\widehat{K}, \widehat{M})$ is much easier to solve than the original eigenvalue problem (K, M) because the number of selected eigenpairs in (18) *is typically quite small and therefore the order of the reduced problem is much smaller than the order of the original problem. If for example the mode selection strategy described in Remark 4 is used then the size of the reduced problem can be bounded by $\mathcal{O}(N^{1/3})$ and the problem can be solved by dense linear algebra routines in $\mathcal{O}(N)$.*

3.3 Multi-Level Version: Algebraic Setting

The single-level version of the AMLS method explained in the previous section can easily be extended to a multi-level version. Using the substructuring from the single-level version 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 version is analogous to the single-level version. As in (17) 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

$$K = \begin{array}{c|ccccccc|} & \Omega_{1,1} & \Omega_{1,2} & \Gamma_1 & \Omega_{2,1} & \Omega_{2,2} & \Gamma_2 & \Gamma \\ \Omega_{1,1} & K_{11} & & K_{13} & & & & K_{17} \\ \Omega_{1,2} & & K_{22} & K_{23} & & & & K_{27} \\ \Gamma_1 & K_{31} & K_{32} & K_{33} & & & & K_{37} \\ \Omega_{2,1} & & & & K_{44} & & K_{46} & K_{47} \\ \Omega_{2,2} & & & & & K_{55} & K_{56} & K_{57} \\ \Gamma_2 & & & & K_{64} & K_{65} & K_{66} & K_{67} \\ \Gamma & K_{71} & K_{72} & K_{73} & K_{74} & K_{75} & K_{76} & K_{77} \end{array} . \tag{21}$$

$K_{ij} \in \mathbb{R}^{N_i \times N_j}$ 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 the substructured domain Ω. Here we want to note that the multi-level version of AMLS does not correspond to a recursive call of the single-level version. Instead the different matrix operations, done in the single-level version, are applied analogously to the matrices from the multi-level version, i.e., to matrices of the form (21) for example.

In the next step the eigenvalue problem (K, M) is transformed equivalently to $(\widetilde{K}, \widetilde{M})$, i.e., K is block diagonalised via $K = L\widetilde{K}L^T$ by performing a block LDL^T-decomposition and M is transformed correspondingly by $\widetilde{M} = L^{-1}ML^{-T}$. Due to the transformation a part of the sparsity structure is lost in $\widetilde{K}$ and $\widetilde{M}$. All submatrices $\widetilde{K}_{ij}$ and $\widetilde{M}_{ij}$ 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 $(\widetilde{K}_{ii}, \widetilde{M}_{ii})$ are computed. Note that $\widetilde{K}_{ii} = K_{ii}$ and $\widetilde{M}_{ii} = M_{ii}$ if their row indices are associated with one of the subdomains. Let the partial eigensolution be given again by

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

for $i = 1, \dots, m$, where the diagonal matrix $D_i \in \mathbb{R}^{k_i \times k_i}$ contains the $k_i \leq N_i$ smallest eigenvalues and $S_i \in \mathbb{R}^{N_i \times k_i}$ column-wise the associated eigenvectors. In the next step the reduced eigenvalue problem $(\widehat{K}, \widehat{M})$ is obtained by computing $\widehat{K} := S^T \widetilde{K} S$ and $\widehat{M} := S^T \widetilde{M} S$ with $S := \mathrm{diag}\big[S_1, \dots, S_m\big]$. 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 (20).

For further illustration we refer to [14] where a two-level version of AMLS in the algebraic setting is described, and for the description of the multi-level version in the continuous setting we refer to [6]. To summarise the AMLS method an overview of all necessary operations is given in Table 1 where the different tasks of the method are denoted by **(T1)**–**(T8)**.

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 (18) the size of the subproblems $(\widetilde{K}_{ii}, \widetilde{M}_{ii})$ is small enough to be solved easily. If more and more levels are used in the multi-level approach of AMLS, then the size of the reduced eigenvalue problem increases as $k = \sum_{i=1}^{m} k_i$ grows with the number m of subproblems. Although the reduced problem is partially structured (the structure is inherited from the block-sparsity of K and M), eventually the total complexity is dominated by this part. As a consequence, the number of levels has to be controlled so that at most $\mathcal{O}(n_{ev})$ eigenvectors are used from all subproblems together. This can be achieved by using only a few levels and recursively applying AMLS for the non-interface subproblems.

3.4 Recursive AMLS

In our new recursive version of (multi-level) AMLS the subdomain eigenvalue problems are solved recursively by the AMLS method. In each of the subdomains Ω_i there are $k_i < n_{ev}$ eigenvectors that can be represented well in the finite element space V_{N_i}. If the number of subdomains (from the multilevel substructuring) is in $\mathcal{O}(1)$ then clearly the size of the reduced problem is at most $\mathcal{O}(n_{ev})$ and can be handled by standard dense linear algebra solvers.

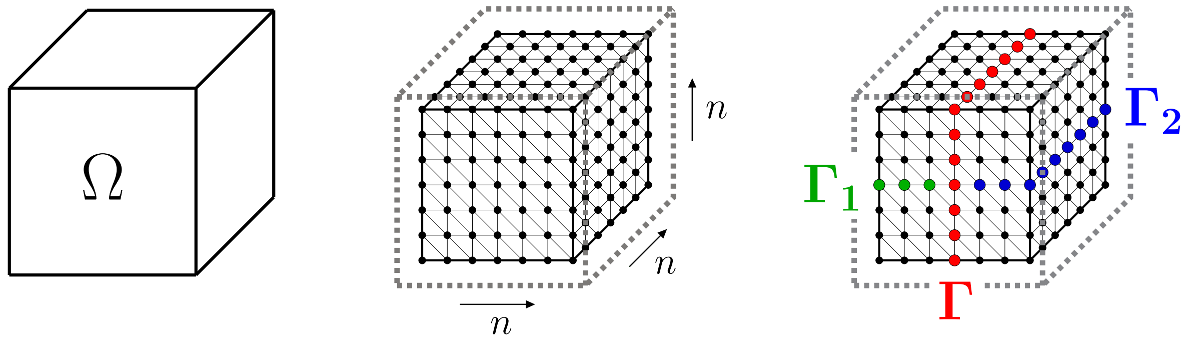
However, neither the recursive approach nor the multi-level approach of AMLS affect the size of interface problems, they affect only the size of subproblems related to subdomains. When the spatial domain Ω is three-dimensional this is a bottleneck.

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 3(b). 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 (21). The number of rows or columns of the interface matrices are $\mathcal{O}(N^{2/3})$ as it is illustrated in Figure 3(c). 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 eigensolution. 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 $\widetilde{K}_{33}$ and $\widetilde{K}_{66}$



(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.

has to be computed when the block LDL^T-decomposition is performed to block diagonalise $K = L\widetilde{K}L^T$. These operations alone lead to costs of $\mathcal{O}\big((N^{2/3})^3\big) = \mathcal{O}\big(N^2\big)$. A way to help out here are so-called hierarchical matrices which are introduced briefly in the next section.

5 Hierarchical Matrices

$\mathcal{H}$-matrices [20, 21] are data-sparse but possibly dense matrices. The underlying idea is to reorder the rows 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 $\mathcal{O}(N \log^\alpha N)$ data is necessary instead of storing N^2 entries where $\alpha = 1, \dots, 4$ (cf. [15, 17]). 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 $O(N \log^\alpha N)$.

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, 13] and [2, 13, 16, 29] 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 $K = L\widetilde{K}L^T$ and the matrix transformation $\widetilde{M} = L^{-1}ML^{-T}$.

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 $i \in I := \{1, \ldots, N\}$ 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

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

Correspondingly each submatrix

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

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*

$$\min\bigl\{\operatorname{diam}(\Omega_s), \operatorname{diam}(\Omega_t)\bigr\} \le \eta \operatorname{dist}(\Omega_s, \Omega_t) \tag{22}$$

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 typically set to $\eta = 1$ (see, e.g., [15]). However, we obtained better results in our numerical tests according to the computational time using larger η and correspondingly having many admissible subblocks. In [22] better results have been obtained as well when even subblocks $s \times t$ with $s \neq t$ were accepted as admissible. In our numerical tests $\eta := 50$ has been a good choice and this value is used in the rest of the paper. The quantities

$$\operatorname{diam}(\Omega_t) := \max\bigl\{\|x-y\|_2 \bigm| x, y \in \Omega_t\bigr\} \quad \text{and} \quad \operatorname{dist}(\Omega_s, \Omega_t) := \min\bigl\{\|x-y\|_2 \bigm| x \in \Omega_s, y \in \Omega_t\bigr\}$$

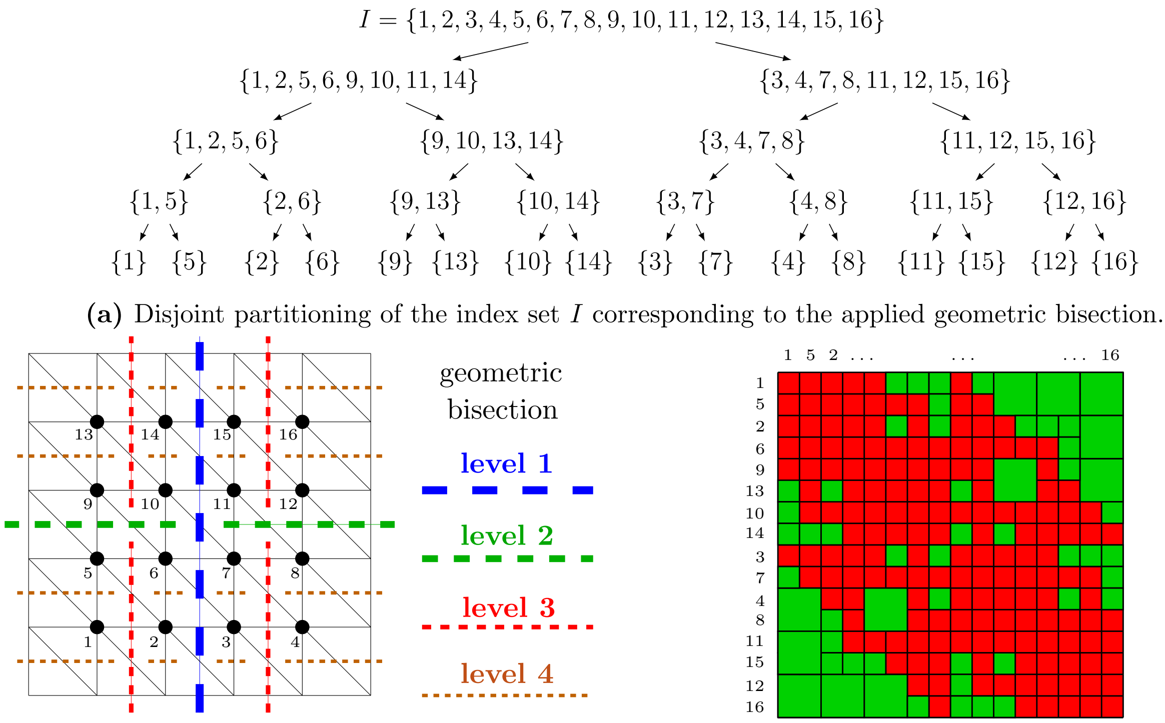
are the diameter and the distance of the supports of s and t. Subblocks $s \times t$ fulfilling the admissibility condition (22) are called *admissible* and the corresponding submatrices $A|_{s\times t}$ are approximated by so-called Rk-matrices which are defined as follows.

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

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

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 $n\,m$. 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 (22) 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 $s, t \subset I$ with $I = s \mathbin{\dot\cup} t$. 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



(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 enumerated from 1 in the lower left to 16 in the upper right corner.

(c) $\mathcal{H}$-Matrix format of $A^{-1} \in \mathbb{R}^{16\times16}$ according to the applied partitioning of I using admissibility condition (22) and $n_{min} = 1$; admissible blocks are coloured green, inadmissible ones are red.

Figure 4: Construction of the $\mathcal{H}$-matrix format 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.

4(a) and 4(b) 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 (22) 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 4(c). 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

$$\frac{\| A^{-1}|_{s\times t} - R \|_2}{\| A^{-1}|_{s\times t} \|_2} \leq \varepsilon \tag{23}$$

where the rank $k \in \mathbb{N}_0$ is as small as possible (cf. [15]).

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 R$k$-matrix with accuracy $\varepsilon$;
    else if $\min\{\#s, \#t\} \leq n_{min}$ then
        represent $A^{-1}|_{s\times t}$ by a full matrix;    ▷ $n_{min}$ affects the minimal size of the submatrices
    else
        $S(s \times t) := \{s' \times t' \mid s' \in S(s),\, t' \in S(t)\}$;    ▷ $S(t)$ 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
```

The $\mathcal{H}$-matrix format and the $\mathcal{H}$-matrix approximation of A^{-1} have been introduced using Algorithm 1. However, this algorithm requires that A^{-1} is explicitly available. Fortunately the $\mathcal{H}$-matrix algebra provides an efficient algorithm, requiring only the matrix A and the used $\mathcal{H}$-matrix format, to compute the $\mathcal{H}$-matrix approximation of A^{-1}. Using a recursive approach applied block-wise to the matrix structure, exploiting the Rk-matrix representation of submatrices fulfilling (22), and applying the inexpensive addition and multiplication of Rk-matrices this algorithm computes the $\mathcal{H}$-matrix approximation of A^{-1} in $\mathcal{O}(N \log^{\alpha} N)$, cf.[15]. Let $(A^{\mathcal{H}})^{-1}$ denote the $\mathcal{H}$-matrix approximation computed by this efficient algorithm. We have to remark that $(A^{\mathcal{H}})^{-1}$ slightly differs from the result obtained by algorithm 1 due to applied approximative matrix operations. Nevertheless, the error $\|A^{-1} - (A^{\mathcal{H}})^{-1}\|$ can be controlled by the chosen accuracy ε in (23).

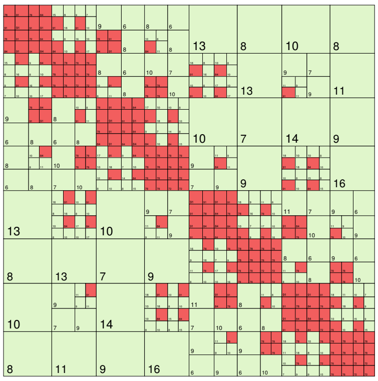


Figure 5: $\mathcal{H}$-matrix format of $A^{-1} \in \mathbb{R}^{N\times N}$ with $N = 2500$ using admissibility condition (22) and $n_{min} = 100$ where A is the stiffness matrix resulting from a finite element discretisation of an elliptic partial differential operator on $\Omega = (0,1)^2$.

To compute the block diagonalisation $K = L\widetilde{K}L^T$ and the matrix transformation $\widetilde{M} = L^{-1}ML^{-T}$ by the fast $\mathcal{H}$-matrix algebra we slightly change the described $\mathcal{H}$-matrix format, cf. [17]. First we apply a nested dissection as in the classical AMLS method, i.e., the domain 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 (17) or (21) is obtained. As discussed in Section 3 some of the submatrices $\widetilde{K}_{ij}$ and $\widetilde{M}_{ij}$ are fully populated, however, they can be

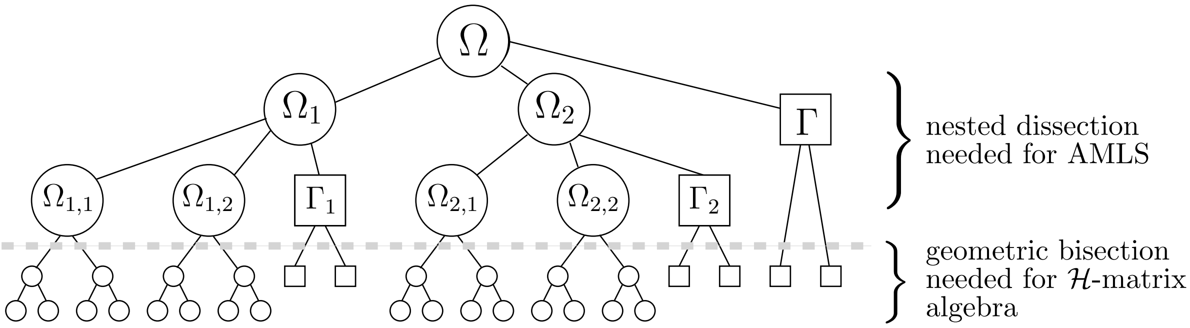


Figure 6: Schematic example of the partitioning of the domain Ω applied in $\mathcal{H}$-AMLS: A two-level nested dissection (necessary for AMLS, cf. Figure 2) is applied followed by an additional two-level geometric bisection of the subdomains and a one-level geometric bisection of the interfaces (necessary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

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 $\widetilde{K}_{ij}$ and $\widetilde{M}_{ij}$ 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 an efficient algorithm, similar to the recursive algorithm used for $(A^{\mathcal{H}})^{-1}$, in $\mathcal{O}(N \log^\alpha N)$ leading to

$$K \approx L^{\mathcal{H}} \widetilde{K}^{\mathcal{H}} (L^{\mathcal{H}})^T \quad \text{and} \quad \widetilde{M}^{\mathcal{H}} \approx (L^{\mathcal{H}})^{-1} M (L^{\mathcal{H}})^{-T}. \tag{24}$$

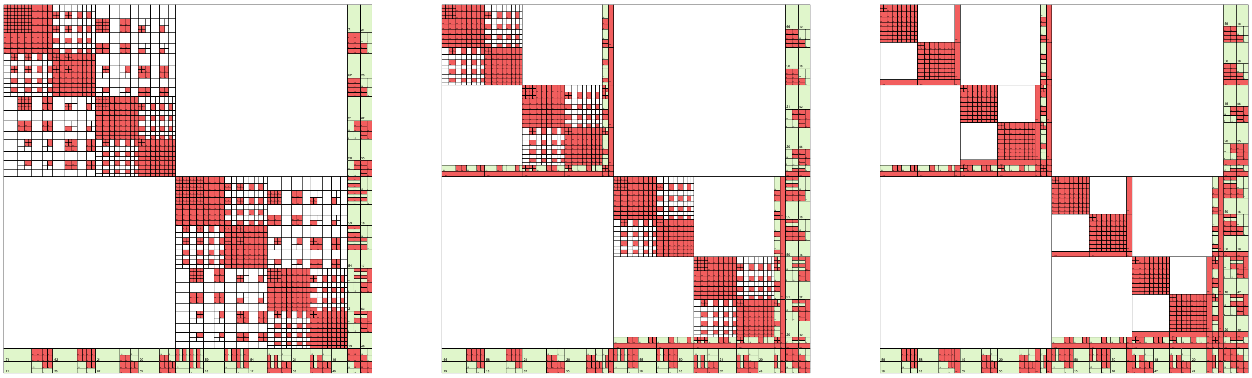
As already noted, these $\mathcal{H}$-matrix operations are performed not exactly but only approximatively, and the approximation errors $\|L - L^{\mathcal{H}}\|_2$, $\|\widetilde{K} - \widetilde{K}^{\mathcal{H}}\|_2$ and $\|\widetilde{M} - \widetilde{M}^{\mathcal{H}}\|_2$ are influenced by the chosen accuracy ε in (23). An example of the applied $\mathcal{H}$-matrix format is given in Figure 7 for the matrix $\widetilde{M}^{\mathcal{H}}$.

6 Combination of AMLS and $\mathcal{H}$-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 (24) 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 $\widetilde{K}^{\mathcal{H}}$ and $\widetilde{M}^{\mathcal{H}}$ according to block row i and block column j are denoted by $\widetilde{K}^{\mathcal{H}}_{ij}$ and



(a) one-level nested dis. **(b)** two-level nested dis. **(c)** three-level nested dis.

Figure 7: $\mathcal{H}$-matrix format 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.

$\widetilde{M}^{\mathcal{H}}_{ij}$. In the next step the partial eigensolutions of the subproblems $(\widetilde{K}^{\mathcal{H}}_{ii}, \widetilde{M}^{\mathcal{H}}_{ii})$ are computed for $i = 1, \ldots, m$ which are given by

$$\widetilde{K}^{\mathcal{H}}_{ii}\mathbf{S}_i \;=\; \widetilde{M}^{\mathcal{H}}_{ii}\mathbf{S}_i\,\mathbf{D}_i \quad \text{with} \quad \mathbf{S}_i^T\widetilde{M}^{\mathcal{H}}_{ii}\mathbf{S}_i = \mathrm{Id}, \tag{25}$$

where the diagonal matrix $\mathbf{D}_i \in \mathbb{R}^{k_i \times k_i}$ contains the $k_i \leq N_i$ smallest eigenvalues and the matrix $\mathbf{S}_i \in \mathbb{R}^{N_i \times k_i}$ column-wise the associated eigenvectors. Because in general the matrices $\widetilde{K}^{\mathcal{H}}_{ii}$ and $\widetilde{M}^{\mathcal{H}}_{ii}$ slightly differ from $\widetilde{K}_{ii}$ and $\widetilde{M}_{ii}$ the corresponding eigensolutions (18) and (25) 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 $\mathbf{S} := \mathrm{diag}\,[\mathbf{S}_1, \ldots, \mathbf{S}_m]$ and the matrices

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

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

$$\begin{cases} \text{find } (\widehat{\boldsymbol{\lambda}}, \widehat{\mathbf{x}}) \in \mathbb{R} \times \mathbb{R}^k \text{ with} \\ \widehat{\mathbf{K}}\,\widehat{\mathbf{x}} \;=\; \widehat{\boldsymbol{\lambda}}\,\widehat{\mathbf{M}}\,\widehat{\mathbf{x}} \end{cases} \tag{26}$$

where the eigenpairs be given by $\big(\widehat{\boldsymbol{\lambda}}_j, \widehat{\mathbf{x}}_j\big)_{j=1}^{k} \in \mathbb{R}_{>0} \times \mathbb{R}^k$ with $\widehat{\boldsymbol{\lambda}}_j \leq \widehat{\boldsymbol{\lambda}}_{j+1}$. In the last step the smallest n_{ev} eigenpairs of (26) are computed leading to the eigenpair approximations $(\widehat{\boldsymbol{\lambda}}_j, \widehat{\mathbf{y}}_j)_{j=1}^{n_{ev}}$ 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 $\widehat{\boldsymbol{\lambda}}_j$ is not equal to the Rayleigh quotient

$$\widehat{\boldsymbol{\lambda}}_j^{rq} := \widehat{\mathbf{y}}_j^T K\,\widehat{\mathbf{y}}_j /\, \widehat{\mathbf{y}}_j^T M\,\widehat{\mathbf{y}}_j \tag{27}$$

since the matrix operations in (24) are performed only approximatively. Typically the Rayleigh quotients $\widehat{\boldsymbol{\lambda}}_j^{rq}$ deliver better approximations of the sought eigenvalues λ_j than $\widehat{\boldsymbol{\lambda}}_j$, 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 denoted by **(T1)**–**(T8)**.

Task	Matrix Operations AMLS	Matrix Operations $\mathcal{H}$-AMLS
(T1) partition matrices K and M	nested dissection reordering, cf. (17) and (21)	nested dissection reordering, cf. (17) and (21), with subsequent geometric bisection (cf. Section 5)
(T2) block diagonalise K	$K = L\widetilde{K}L^T$ → **expensive** because of large-sized, dense interface matrices	$K \approx L^{\mathcal{H}}\widetilde{K}^{\mathcal{H}}(L^{\mathcal{H}})^T$ → using **fast $\mathcal{H}$-matrix algebra** done in $\mathcal{O}(N\log^\alpha N)$
(T3) transform M	$\widetilde{M} = L^{-1}ML^{-T}$ → **expensive** because of large-sized, dense interface matrices	$\widetilde{M}^{\mathcal{H}} \approx (L^{\mathcal{H}})^{-1}M(L^{\mathcal{H}})^{-T}$ → using **fast $\mathcal{H}$-matrix algebra** done in $\mathcal{O}(N\log^\alpha N)$
(T4) compute partial eigensolution for $i = 1,\ldots,m$	$\widetilde{K}_{ii}S_i = \widetilde{M}_{ii}S_iD_i$ → **expensive** when $\widetilde{K}_{ii}$ and $\widetilde{M}_{ii}$ are interface matrices because they are dense	$\widetilde{K}^{\mathcal{H}}_{ii}\mathbf{S}_i = \widetilde{M}^{\mathcal{H}}_{ii}\mathbf{S}_i\mathbf{D}_i$ → use **fast $\mathcal{H}$-matrix algebra** when $\widetilde{K}^{\mathcal{H}}_{ii}$ and $\widetilde{M}^{\mathcal{H}}_{ii}$ are interface matrices
(T5) define subspace	$S := \operatorname{diag}(S_1,\ldots,S_m) \in \mathbb{R}^{N\times k}$ with $k = \sum_{i=1}^m k_i$	$\mathbf{S} := \operatorname{diag}(\mathbf{S}_1,\ldots,\mathbf{S}_m) \in \mathbb{R}^{N\times k}$ with $k = \sum_{i=1}^m k_i$
(T6) compute matrices of reduced eigenvalue problem	$\widehat{K} := S^T\widetilde{K}S \in \mathbb{R}^{k\times k}$, $\widehat{M} := S^T\widetilde{M}S \in \mathbb{R}^{k\times k}$	$\widehat{\mathbf{K}} := \mathbf{S}^T\widetilde{K}^{\mathcal{H}}\mathbf{S} \in \mathbb{R}^{k\times k}$, $\widehat{\mathbf{M}} := \mathbf{S}^T\widetilde{M}^{\mathcal{H}}\mathbf{S} \in \mathbb{R}^{k\times k}$ → use **fast $\mathcal{H}$-matrix algebra** for computation
(T7) solve reduced eigenvalue problem	$\widehat{K}\,\widehat{x}_j = \widehat{\lambda}_j\,\widehat{M}\,\widehat{x}_j$ for $j = 1,\ldots,n_{ev}$	$\widehat{\mathbf{K}}\,\widehat{\mathbf{x}}_j = \widehat{\boldsymbol{\lambda}}_j\,\widehat{\mathbf{M}}\,\widehat{\mathbf{x}}_j$ for $j = 1,\ldots,n_{ev}$
(T8) transformation of eigenvectors	$\widehat{y}_j := L^{-T}S\,\widehat{x}_j$ for $j = 1,\ldots,n_{ev}$	$\widehat{\mathbf{y}}_j := (L^{\mathcal{H}})^{-T}\mathbf{S}\,\widehat{\mathbf{x}}_j$ for $j = 1,\ldots,n_{ev}$
final eigenpair approximations	$(\widehat{\lambda}_j,\widehat{y}_j)$ for $j = 1,\ldots,n_{ev}$	$(\widehat{\boldsymbol{\lambda}}^{rq}_j,\widehat{\mathbf{y}}_j)$ for $j = 1,\ldots,n_{ev}$ with $\widehat{\boldsymbol{\lambda}}^{rq}_j := \widehat{\mathbf{y}}_j^T K\,\widehat{\mathbf{y}}_j \,/\, \widehat{\mathbf{y}}_j^T M\,\widehat{\mathbf{y}}_j$

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 (24) and the applied modal truncation in (25), i.e, the number of selected eigenvectors k_i. A coarser accuracy ε and smaller k_i result in faster computations and reduced memory requirements of $\mathcal{H}$-AMLS. Of course these parameters can be chosen arbitrarily, however, their choice influences the 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, $\mathcal{O}(N\log N)$. 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)**) is faster in the $\mathcal{H}$-AMLS method: The submatrices $\widetilde{K}^{\mathcal{H}}_{ii}$ and $\widetilde{M}^{\mathcal{H}}_{ii}$ 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 $\mathcal{H}$-matrix structure can be applied in (25) instead of an eigensolver for dense matrices as it is done in classical AMLS. Since the interface matrices are of size at most $\mathcal{O}(N^{2/3})$, the almost linear scaling of $\mathcal{H}$-matrices allows us to solve for n_{ev} eigenvectors in complexity $\mathcal{O}\big(n_{ev}N^{2/3}(\log^{\alpha} N + n_{ev})\big)$, e.g. by SIL, which is for $n_{ev} \sim N^{1/3}$ the same as $\mathcal{O}(n_{ev}N)$.

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 $\widehat{\mathbf{M}}$: The multiplications $\mathbf{S}^T(\widetilde{K}^{\mathcal{H}}\mathbf{S})$ and $\mathbf{S}^T(\widetilde{M}^{\mathcal{H}}\mathbf{S})$ involve $2n_{ev}$ $\mathcal{H}$-matrix times vector multiplications in $\mathcal{O}(N\log^{\alpha} N)$ plus $2n_{ev}^2$ scalar products of length N. Both together sum up to costs of the order $\mathcal{O}\big(n_{ev}N(\log^{\alpha} N + n_{ev})\big)$ for task **(T6)**. Since the scalar products can be computed with peak performance, the costs for these is invisible in practice. The n_{ev} $\mathcal{H}$-matrix times vector multiplications are as well harmless since the logarithms and constants involved in the matrix vector multiplications are much smaller than for the $\mathcal{H}$-matrix operations in (24).

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 of the same structure. Since we aim at $n_{ev} \sim N^{1/3}$ eigenvalues, the size of the reduced problem allows us to use a dense linear algebra solver with cubic complexity and still remain in $\mathcal{O}(N)$.

Finally, for task **(T8)** we can again exploit the fast $\mathcal{H}$-matrix times vector multiplication (forward substitution in $(L^{\mathcal{H}})^T$) to complete this task in $\mathcal{O}\big(n_{ev}N(\log^{\alpha} N + n_{ev})\big)$ and for the Rayleigh Quotients it is enough to use the sparsity of K and M to perform the computation in $\mathcal{O}(n_{ev}N)$.

We can sum up that theoretically the complexity is dominated by tasks **(T6)** and **(T8)**. The operations involved there are the $\mathcal{H}$-matrix times vector multiplication, which accumulates to a total of $\mathcal{O}(n_{ev}N\log^{\alpha} N)$, and the usual scalar product accumulating to at most $\mathcal{O}(n_{ev}^2N)$ multiplications or additions. Both of these operations have extremely small constants involved and are therefore for problem sizes up to $N = 4,000,000$ not the bottleneck. Instead, most of the time is spent in the transformation steps **(T2)** and **(T3)**, both of them in $\mathcal{O}(N\log^{\alpha} N)$ which is asymptotically in $o(n_{ev}N)$, and in the interface eigenvalue problem of task **(T4)** which is of complexity $\mathcal{O}(n_{ev}N)$.

In the numerical examples we can observe that the costs for **(T6)** and **(T8)** are slowly increasing relative to the total cost, and that the total complexity stays in $\mathcal{O}(n_{ev}N)$ for very large-scale problems.

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 number of selected eigenvectors k_i — is a possible loss in quality of the eigenpair approximations.

Keeping in mind the initial problem, the Rayleigh quotients $\widehat{\boldsymbol{\lambda}}_j^{rq}$ in (27) are used to approximate the n_{ev} smallest eigenvalues λ_j of the continuous problem (2). For the approximation error it holds

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

where $\lambda_j^{(N)}$ is the eigenvalue of the discrete problem (4) and $\widehat{\lambda}_j$ is the eigenvalue of the reduced problem (19) from classical AMLS. The upper index of $\lambda_j^{(N)}$ is indicating the number of 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 number of selected eigenvectors k_i in (25), and the error caused by the use of $\mathcal{H}$-matrix approximation is influenced by the chosen accuracy ε in (24).

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 (almost) exact eigenvalues $\lambda_j^{(N)}$ of the discrete problem (4) are computed. The corresponding discretisation errors are used as 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

$$\underbrace{|\lambda_j - \widehat{\boldsymbol{\lambda}}_j^{rq}|}_{\text{error of }\mathcal{H}\text{-AMLS}} \quad\approx\quad \underbrace{|\lambda_j - \lambda_j^{(N)}|.}_{\text{discretisation error}} \tag{28}$$

Dividing (28) by $|\lambda_j|$ we obtain the equivalent statement expressed in form of relative errors

$$\widehat{\boldsymbol{\delta}}_j^{rq} := \underbrace{\frac{|\lambda_j - \widehat{\boldsymbol{\lambda}}_j^{rq}|}{|\lambda_j|}}_{\substack{\text{relative error}\\ \text{of }\mathcal{H}\text{-AMLS}}} \quad\approx\quad \underbrace{\frac{|\lambda_j - \lambda_j^{(N)}|}{|\lambda_j|}}_{\substack{\text{relative error}\\ \text{of discretisation}}} =: \delta_j^{(N)}. \tag{29}$$

In the following the aim is to choose the parameters k_i and ε in such a way that (29) holds while the computational costs and storage requirements of $\mathcal{H}$-AMLS are reduced as much as possible.

7 Numerical Results

The $\mathcal{H}$-AMLS method has been implemented in C++ using the $\mathcal{H}$-matrix software library HLIBpro [7, 26, 27]. In the following we analyse numerically $\mathcal{H}$-AMLS for the Laplace eigenvalue problem

$$\begin{cases} -\Delta u = \lambda u & \text{in } \Omega = (0,1)^3, \\ \quad\;\; u = 0 & \text{on } \partial\Omega. \end{cases} \tag{30}$$

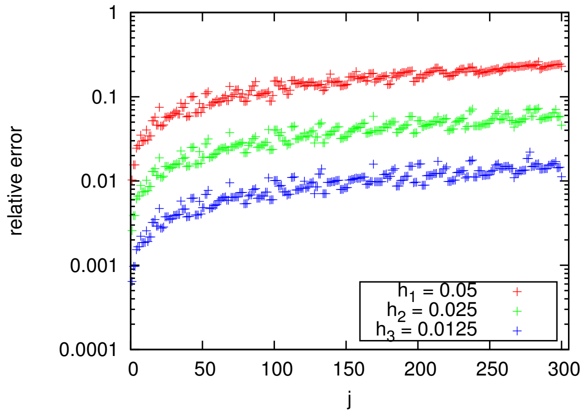


Figure 8: Relative discretisation errors $\delta_j^{(N)}$ of the smallest 300 eigenvalues of the problem (30) for varying mesh widths h.

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 (30) 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 $\widehat{\boldsymbol{\delta}}_j^{rq}$ from (29).

To solve problem (30) 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 is the number of 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 (30). 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

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

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 $\delta_j^{(N)}$ form more or less a monotonically 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 iteration[1] where the arising shift-invert systems have been solved with the help of the fast $\mathcal{H}$-matrix algebra — but any other solver would suffice.

Neglecting possible computational costs for the orthogonalisation and the predonditioners, however, a lower bound for the best possible computational complexity of an eigensolver

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

j	λ_j	**error** $\lvert\lambda_j - \lambda_j^{(N)}\rvert$			**rel. error** $\delta_j^{(N)}$			$\max\{\,\delta_i^{(N)} \mid i = 1, \ldots, j\,\}$		
		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 (30) 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.)

would be

$$\mathcal{O}\big(\, n_{ev}\, N \,\big). \tag{31}$$

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

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

Assume for example that a classical iterative approach has the best possible complexity where in average 10 iterations are necessary until an iteration vector converges, and the matrix-vector multiplication (by the inverse) takes 5 seconds per one million DOF. Then the average computational time of this eigensolver is $\operatorname{avg}(t_{\text{all}}) = 50s$.

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 (26) is obtained where the Rayleigh quotients $\widehat{\boldsymbol{\lambda}}_j^{rq}$ are approximating the sought eigenvalues λ_j of (30). Beside the DOF of the model, the relative errors $\widehat{\boldsymbol{\delta}}_j^{rq}$ depend on the number of selected eigenvectors k_i 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 match the discretisation errors. In particular we will test for $n_{ev} = N^{1/3}, 2N^{1/3}, 5N^{1/3}$ how the parameters have to be selected so that the inequality

$$\gamma_{n_{ev}}^{(N)} < 3 \tag{33}$$

holds where

$$\gamma_{n_{ev}}^{(N)} := \max\Big\{ \widehat{\boldsymbol{\delta}}_j^{rq} \,/\, \widehat{\delta}_j^{(N)} \mid j = 1, \ldots, n_{ev} \Big\}$$

is the maximal ratio between the relative discretisation error $\delta_j^{(N)}$ and the relative error $\widehat{\boldsymbol{\delta}}_j^{rq}$ associated to $\mathcal{H}$-AMLS. If inequality (33) is fulfilled it can be said that the approximation error of $\mathcal{H}$-AMLS is of the same order as the discretisation error, cf. (29).

7.1 Influence of the Modal Truncation

At first we investigate the influence of the number of selected eigenvectors k_i. To do this the $\mathcal{H}$-matrix approximation is deactivated in (24) by setting the parameter η from (22) 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 (24) 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.

We used the approach discussed in Remark 4 for the modal truncation and benchmarked the following two mode selection strategies:

strategy	subdomain problem	interface problem
• SI	$k_i = 1.5 N_i^{1/3}$	$k_i = N_i^{1/3}$
• SII	$k_i = 1.5 N_i^{1/3}$	$k_i = N_i^{1/2}$

If for example strategy SII is applied then the smallest $k_i = 1.5 N_i^{1/3}$ eigenpairs in (25) are selected if the corresponding subproblem is associated to a subdomain and the smallest $k_i = N_i^{1/2}$ eigenpairs if the subproblem is associated to an interface. In Figure 9(a) the corresponding relative errors $\widehat{\boldsymbol{\delta}}_j^{rq}$ and in Table 3 the maximal ratios $\gamma_{n_{ev}}^{(N)}$ are displayed for the mesh widths h_1, h_2 and h_3. For comparison the discretisation errors $\widehat{\delta}_j^{(N)}$ are displayed as well in Figure 9(a). Obviously, strategy SI (where only $N_i^{1/3}$ modes from the interface are selected) deteriorates as $h \to 0$.

In Table 3 can be seen that for h_1,h_2 and h_3 mode selection strategy SII is sufficient in such a way that for all $n_{ev} = N^{1/3}, 2N^{1/3}, 5N^{1/3}$ postulation (33) is fulfilled. However, the computational costs of $\mathcal{H}$-AMLS are getting very expensive with increasing DOF because $\eta = 0$.

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 (24) the $\mathcal{H}$-matrix approximation is activated by setting the parameter η in (22) back to $\eta = 50$. Accordingly certain subblocks get admissible and the respective submatrices are approximated by Rk-matrices with a given approximation accuracy ε.

In the previous subsection we have seen that mode selection strategy SII is sufficient for the mesh widths h_1, h_2 and h_3. Using this mode selection strategy the computations have been done again applying the following $\mathcal{H}$-matrix approximation accuracies

- $\varepsilon_1 := \varepsilon_1(h) := 6 \cdot h$
- $\varepsilon_2 := \varepsilon_2(h) := 120 \cdot h^2$.

The accuracies depend on the mesh width of the underlying model and for h_1, h_2, h_3 we obtain:

	h_1	h_2	h_3
$\varepsilon_1(h)$	0.3	0.15	0.075
$\varepsilon_2(h)$	0.3	0.075	0.01875

n_{ev}	$\gamma_{n_{ev}}^{(N)}$ for h_1		$\gamma_{n_{ev}}^{(N)}$ for h_2		$\gamma_{n_{ev}}^{(N)}$ for h_3	
	SI	SII	SI	SII	SI	SII
$N^{1/3}$	4.93	1.78	7.49	1.79	21.47	1.91
$2N^{1/3}$	5.00	1.78	7.49	2.02	21.47	2.18
$5N^{1/3}$	5.00	2.58	8.36	2.06	21.47	2.37

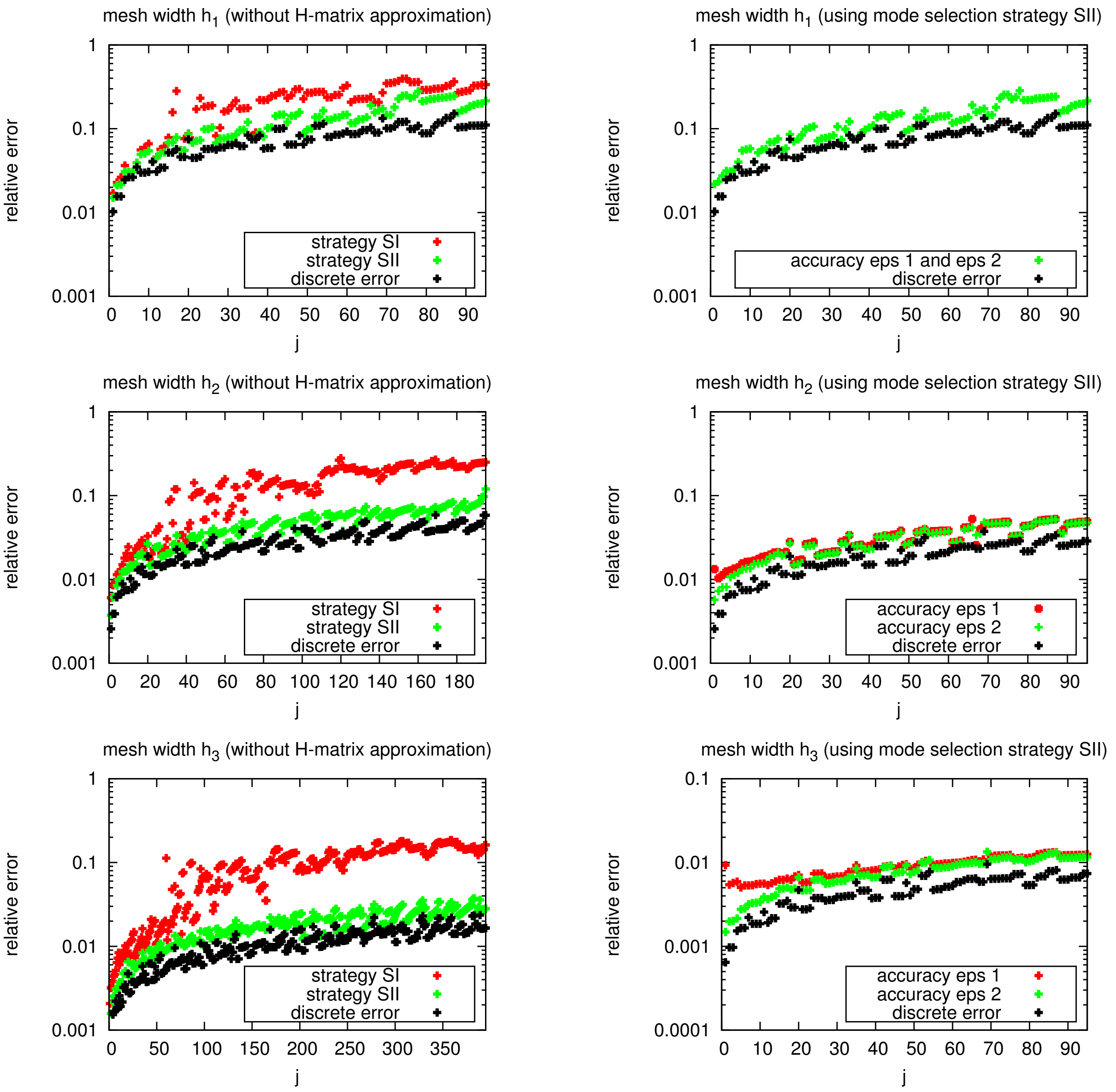
Table 3: Influence of the mode selection strategy to the maximal ratios $\gamma_{n_{ev}}^{(N)}$ for varying mesh widths. In this test the $\mathcal{H}$-matrix approximation has been deactivated (η was set to 0).

n_{ev}	$\gamma_{n_{ev}}^{(N)}$ for h_1			$\gamma_{n_{ev}}^{(N)}$ for h_2			$\gamma_{n_{ev}}^{(N)}$ for h_3		
	ε_1	ε_2	$\eta = 0$	ε_1	ε_2	$\eta = 0$	ε_1	ε_2	$\eta = 0$
$N^{1/3}$	2.11	2.11	1.78	5.15	2.22	1.79	14.46	2.31	1.91
$2N^{1/3}$	2.11	2.11	1.78	5.15	2.22	2.02	14.46	2.31	2.18
$5N^{1/3}$	2.63	2.63	2.58	5.15	2.22	2.06	14.46	2.40	2.37

Table 4: Influence of the $\mathcal{H}$-matrix approximation accuracy $\varepsilon = \varepsilon_1(h), \varepsilon_2(h)$ on the maximal ratios $\gamma_{n_{ev}}^{(N)}$ for varying mesh widths. In this test mode selection strategy SII has been applied.

The relative errors $\widehat{\boldsymbol{\delta}}_j^{rq}$ of this benchmark are displayed in Figure 9(b) and the maximal ratios $\gamma_{n_{ev}}^{(N)}$ in Table 4. In Table 4 can be seen that for the mesh widths h_1, h_2 and h_3 mode selection strategy SII and $\mathcal{H}$-matrix accuracy ε_2 are sufficient to fulfil postulation (33) for all $n_{ev} = N^{1/3}, 2N^{1/3}, 5N^{1/3}$. This parameter setting adjusts the $\mathcal{H}$-matrix accuracy automatically to the underlying mesh width, and the number of selected eigenpairs k_i automatically to the size of the subproblem. Furthermore, in Figure 9(b) can be seen that the approximation of smaller eigenvalues behaves more sensitive to the chosen accuracy ε than the approximation of larger eigenvalues.

We could observe in our benchmarks that the number of selected eigenpairs k_i in (25) should be of the order $\mathcal{O}\big(N_i^{1/3}\big)$ for subdomain problems (which are associated to three-dimensional subdomains) and of the order $\mathcal{O}\big(N_i^{1/2}\big)$ for interface eigenvalue problems (which are associated to hyperplanes in $\mathbb{R}^3$). The accuracy of the $\mathcal{H}$-matrix algebra in (24) should be proportional to h^2, or respectively, expressed in DOF to $N^{-2/3}$. We recommend this parameter setting for similar problems. If more accuracy of the eigenpair approximations is needed k_i should be scaled by a constant larger than 1 and ε_2 by a constant smaller than 1.



(a) Influence of the mode selection strategy to the relative errors $\widehat{\boldsymbol{\delta}}_j^{rq}$. In this tests the $\mathcal{H}$-matrix approximation has been deactivated (η was set to 0). The approximation errors of the smallest $5N^{1/3}$ eigenvalues are displayed.

(b) Influence of the $\mathcal{H}$-matrix approximation accuracy ε to the relative errors $\widehat{\boldsymbol{\delta}}_j^{rq}$. In this tests mode selection strategy SII has been applied. To highlight the influence of the $\mathcal{H}$-matrix accuracy on the approximation error of the smallest eigenvalues only the first 95 are displayed.

Figure 9: Influence of the mode selection strategy and the $\mathcal{H}$-matrix approximation accuracy ε to the relative errors $\widehat{\boldsymbol{\delta}}_j^{rq}$ of $\mathcal{H}$-AMLS for varying mesh widths and comparison with the relative discretisation errors $\delta_j^{(N)}$.

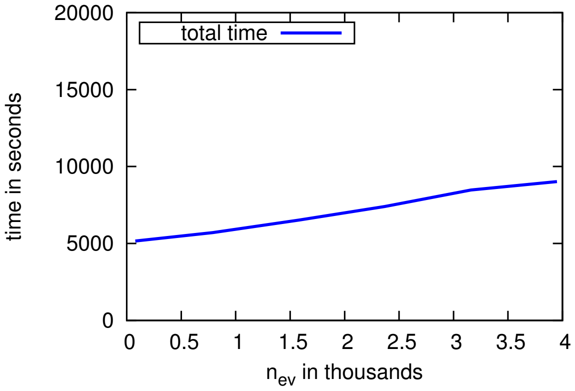


Figure 10: Total computational time of $\mathcal{H}$-AMLS for computing the smallest n_{ev} eigenpairs for the mesh width h_3 using mode selection strategy SII and $\mathcal{H}$-matrix accuracy ε_2. n_{ev} is varying from $N^{1/3} = 79$ up to $50N^{1/3} = 3950$.

7.3 Timing for Multi-Level (non-recursive) $\mathcal{H}$-AMLS

The computational costs of the benchmarks from the previous section, using mode selection strategy SII and $\mathcal{H}$-matrix accuracy ε_2, are given in Table 5 for $n_{ev} = 5N^{1/3}$. 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 $K \approx L^{\mathcal{H}} \widetilde{K}^{\mathcal{H}} (L^{\mathcal{H}})^T$ and the matrix transformation $\widetilde{M}^{\mathcal{H}} \approx (L^{\mathcal{H}})^{-1} M (L^{\mathcal{H}})^{-T}$, task **(T2)** and **(T3)**, are dominating the costs of the other tasks. However, with increasing DOF the portion of task **(T2)** and **(T3)** to the total computational time is decreasing. To keep the computational costs of task **(T4)** small the domain Ω has been substructured several times in our benchmarks to obtain small subdomain eigenvalue problems in (25) which can be solved easily. In order to keep their size constant when h is decreased, the number of levels in $\mathcal{H}$-AMLS has to increase, cf. Figure 6 and column (lvl) in Table 5. The downside of the multi-level substructuring is that for constant sized subdomain eigenvalue problems, the size of the $\mathcal{H}$-reduced eigenvalue problem is $\mathcal{O}(N)$ with the same block-sparsity structure as the original discrete eigenvalue problem. Nevertheless, the cost savings achieved in task **(T4)** outweigh the additional computational costs in tasks **(T6)**–**(T8)**. The eigenpairs of the $\mathcal{H}$-reduced problem have been computed by the dense eigensolver `dsygvx` of LAPACK, and correspondingly in Table 5 we observe that the computational costs of task **(T7)** are increasing much stronger than the costs of tasks **(T6)** and **(T8)**. This is due to the fact that we have not yet applied the recursive version of $\mathcal{H}$-AMLS.

The mode selection strategy and the $\mathcal{H}$-matrix accuracy have been chosen in such a way that postulation (33) is fulfilled, i.e., that the eigenvalue approximation error due to $\mathcal{H}$-AMLS matches the discretisation error. In Table 4 we see that the ratios between $\widehat{\boldsymbol{\delta}}_j^{rq}$ and $\widehat{\delta}_j^{(N)}$ are only slowly increasing in j. It seems that in these benchmarks much more than $5N^{1/3}$ eigenvalue approximations can be computed with nearly the same approximation quality as the discretisation. Increasing the number of sought eigenpairs, however, increases just slightly the computational costs of $\mathcal{H}$-AMLS as it can be seen in Figure 10. There the total computational time of $\mathcal{H}$-AMLS depending on n_{ev} is displayed with n_{ev} up to $50N^{1/3}$.

	characteristics $\mathcal{H}$-AMLS					**computational time of tasks in relation to total time**					**computational time**	
	n_{ev}	N	k	lvl	$\gamma_{n_{ev}}^{(N)}$	**(T2)+(T3)**	**(T4)**	**(T6)**	**(T7)**	**(T8)**	t_{all}	avg(t_{all})
h_1	95	6,859	185	3	2.63	55.1%	41.2%	0.74%	0.31%	2.45%	10s	15.18s
h_2	195	59,319	1,649	6	2.22	69.0%	19.2%	3.07%	3.18%	5.28%	2min 51s	14.86s
h_3	395	493,039	13,537	9	2.40	33.2%	5.24%	7.99%	45.0%	8.38%	1h 27min 44s	27.02s

Table 5: Computational costs of $\mathcal{H}$-AMLS computing the smallest $n_{ev} = 5N^{1/3}$ eigenpairs for varying mesh widths using mode selection strategy SII and $\mathcal{H}$-matrix accuracy ε_2. The computational costs of the tasks **(T1)** and **(T5)** are negligible and left out in this table. t_{all} is the total computational time and avg(t_{all}) the average time defined in (32) using $n_{ev} = 5N^{1/3}$.

	characteristics <u>recursive</u> $\mathcal{H}$-AMLS				**computational time of tasks in relation to total time**					**computational time**	
	n_{ev}	N	k	$\gamma_{n_{ev}}^{(N)}$	**(T2)+(T3)**	**(T4)**	**(T6)**	**(T7)**	**(T8)**	t_{all}	avg(t_{all})
h_1	95	6,859	185	2.63	55.6%	40.6%	0.73%	0.30%	2.47%	10s	15.31s
h_2	195	59,319	1,505	2.22	64.8%	18.2%	6.31%	2.40%	8.10%	3min 5s	16.01s
h_3	395	493,039	3,105	2.99	62.7%	9.92%	11.9%	1.67%	13.6%	46min 40s	14.37s
h_4	795	4,019,679	6,323	($\star$)	47.3%	5.06%	23.0%	1.11%	23.3%	20h 18min 20s	22.87s

Table 6: Computational costs of <u>recursive</u> $\mathcal{H}$-AMLS computing the smallest $n_{ev} = 5N^{1/3}$ eigenpairs for varying mesh widths using the parameter setting described in Section 7.4. The discretisation error ($\star$) was beyond our computing capabilities.

7.4 Timing for Recursive $\mathcal{H}$-AMLS

In the previous section we have seen that the fast $\mathcal{H}$-matrix algebra can be successfully applied for the computation of the transformed eigenvalue problem, task **(T2)** and **(T3)**, leading to a massive reduction of the computational time of AMLS. These two tasks are the bottleneck of classical AMLS, each with costs of at least the order $\mathcal{O}(N^2)$ for three-dimensional problems (cf. Section 4). Using the fast $\mathcal{H}$-matrix algebra these tasks are now computed in almost linear complexity $\mathcal{O}(N \log^\alpha N)$ independently of the number n_{ev} of sought eigenvectors.

In this section we consider the recursive $\mathcal{H}$-AMLS version where the size of the $\mathcal{H}$-reduced eigenvalue problem can be bounded by $\mathcal{O}(N^{1/3})$, cf. Remark 5.

We choose the parameters for recursive $\mathcal{H}$-AMLS as follows:

1. In the multi-level substructuring we use 4 levels accumulating to a total of $m = 31$ subproblems (16 subdomain and 15 interface eigenvalue problems). For the computation of the transformation (24) we use $\mathcal{H}$-matrix accuracy $\varepsilon_2(h)$ from the previous section.

2. For large-sized subdomain problems we compute recursively by $\mathcal{H}$-AMLS the smallest $5N_i^{1/3}$ eigenpairs.

3. For large-sized interface problems we compute the smallest $N_i^{1/2}$ eigenpairs by SIL where the inverse is approximated in the $\mathcal{H}$-matrix format using the fast $\mathcal{H}$-algebra.

4. For small-sized subproblems we use the LAPACK solver `dsygvx` and compute the eigenpairs directly. For subdomain problems we compute the smallest $1.5N_i^{1/3}$ eigenpairs and for interface problems the smallest $N_i^{1/2}$.

Using this parameter setting we achieved that $\gamma_{n_{ev}}^{(N)} < 3$ holds for $n_{ev} = 5N^{1/3}$ and the mesh widths h_1, h_2 and h_3 as it can be seen in Table 6. Analogously to (33), the value $\gamma_{n_{ev}}^{(N)}$ is the maximal ratio between the relative discretisation error and the relative error associated to recursive $\mathcal{H}$-AMLS. Furthermore, in Table 6 we display the computational costs of the different tasks and the size of the $\mathcal{H}$-reduced problem.

We remark that in Table 6 and in the following the time measurements concerning the different tasks have to be seen accumulatively, e.g., the computational time of task **(T7)** in Table 6 includes as well the time spent for the solution of the $\mathcal{H}$-reduced eigenvalue problems of the recursive calls of $\mathcal{H}$-AMLS. For the finest mesh width $h_4 := h_3/2$ we obtain a discrete eigenvalue problem with roughly 4 million DOF.

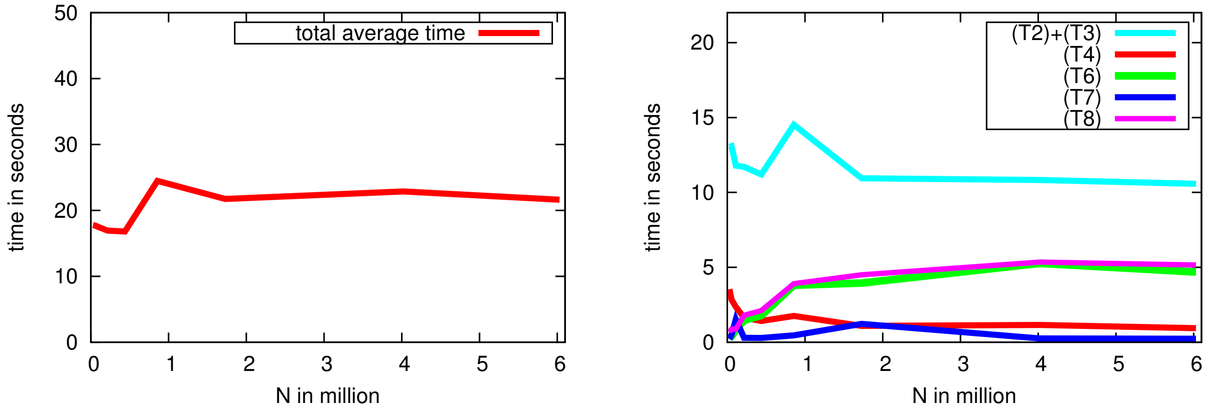
To get a better impression of the practical performance of recursive $\mathcal{H}$-AMLS we investigate the average computational time of the method defined in (32). In Figure 11(a) the average time $\operatorname{avg}(t_{\text{all}})$ is displayed for the computation of the smallest $n_{ev} = 5N^{1/3}$ eigenpairs for varying DOF with N up to 6 million. It can be observed that the average time is constant. Correspondingly recursive $\mathcal{H}$-AMLS reaches in our benchmarks optimal complexity $\mathcal{O}(n_{ev}N)$.

In order to profile the complexity of tasks **(T2)**–**(T8)** (except for **(T5)**) in more detail, we measure the average time for each task separately (accumulated as explained above). The results in Figure 11(b) show that for all involved tasks the average time (per eigenvector) is roughly constant.

8 Conclusion

In order to solve an elliptic PDE eigenvalue problem we have combined a recursive version of 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 $(\widetilde{K}, \widetilde{M})$ 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 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 $(\widetilde{K}_{ii}, \widetilde{M}_{ii})$ and the reduced eigenvalue problem $(\widehat{K}, \widehat{M})$ are performed much more efficiently using the fast $\mathcal{H}$-matrix algebra and the new recursive AMLS. 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.



(a) Total average computational time avg(t_{all}). **(b)** Average computational time of the different tasks.

Figure 11: Average computational time (per eigenvector) of <u>recursive</u> $\mathcal{H}$-AMLS for the computation of the smallest $n_{ev} = 5N^{1/3}$ eigenpairs.

References

[1] 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.

[2] M. Bebendorf. Hierarchical *LU* decomposition based preconditioners for BEM. *Computing*, 74:225–247, 2005.

[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. Börm, L. Grasedyck, and W. Hackbusch. Introduction to hierarchical matrices with applications. *Engineering Analysis with Boundary Elements*, 27(5):405 – 422, 2003.

[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] F. Bourquin and F. d'Hennezel. Numerical study of an intrinsic component mode synthesis method. *Comput. Methods Appl. Mech. Engrg.*, 97(1):49–76, 1992.

[11] R. R. Craig and M. C. C. Bampton. Coupling of substructures for dynamic analysis. *AIAA Journal*, 6:1313–1319, 1968.

[12] K. Elssel and H. Voss. An a priori bound for automated multilevel substructuring. *SIAM J. Matrix Anal. Appl.*, 28(2):386–397 (electronic), 2006.

[13] M. Faustmann, J. Markus Melenk, and D. Praetorius. $\mathcal{H}$-matrix approximability of the inverses of FEM matrices. *ArXiv e-prints*, Aug. 2013.

[14] 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.

[15] L. Grasedyck and W. Hackbusch. Construction and arithmetics of $\mathcal{H}$-matrices. *Computing*, 70(4):295–334, 2003.

[16] 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.

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

[18] 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.

[19] W. Hackbusch. *Elliptic differential equations : theory and numerical treatment*, volume 18 of *Springer series in computational mathematics.* Springer, Berlin, 1992.

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

[21] W. Hackbusch. *Hierarchische Matrizen : Algorithmen und Analysis.* Springer, Dordrecht, 2009.

[22] W. Hackbusch, B. N. Khoromskij, and R. Kriemann. Hierarchical matrices based on a weak admissibility criterion. *Computing*, 73(3):207–243, 2004.

[23] 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.

[24] W. C. Hurty. Vibrations of structural systems by component-mode synthesis. *Journal of the Engineering Mechanics Division*, 86:51–69, 1960.

[25] M. F. Kaplan. *Implementation of automated multi-level substructuring for frequency response analysis of structures.* Ph.d. thesis, Universtiy of Texas at Austin, 2001.

[26] R. Kriemann. HLIBpro. `http://www.hlibpro.com/`.

[27] R. Kriemann. Parallel $\mathcal{H}$-matrix arithmetics on shared memory systems. *Computing*, 74(3):273–297, 2005.

[28] 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.

[29] M. Lintner. The eigenvalue problem for the 2d laplacian in h-matrix arithmetic and application to the heat and wave equation. *Computing*, 72(3-4):293–323, May 2004.

[30] 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.

[31] S. Sauter. hp-finite elements for elliptic eigenvalue problems: Error estimates which are explicit with respect to lambda, h, and p. *SIAM J. Numerical Analysis*, 48(1):95–108, 2010.

[32] P. Seshu. Substructuring and component mode synthesis. *Shock and Vibration*, 4:199–210, 1997.

[33] 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.