PREPRINT 473

DECEMBER 2017

Solving an Integral Equation Eigenvalue Problem via a New Domain Decomposition Method and Hierarchical Matrices

Peter Gerds*

Institut für Geometrie und Praktische Mathematik
Templergraben 55, 52062 Aachen, Germany

* Institut für Geometrie und praktische Mathematik, RWTH Aachen University, Templergraben 55, 52056 Aachen, Germany. Email: gerds@igpm.rwth-aachen.de

Solving an Integral Equation Eigenvalue Problem via a New Domain Decomposition Method and Hierarchical Matrices

Automated Multi-Level Substructuring for Dense Eigenvalue Problems

Peter Gerds*

In this paper the author introduces a new domain decomposition method for the solution of discretised integral equation eigenvalue problems. The new domain decomposition method is motivated by the so-called *automated multi-level substructuring* (short AMLS) method. The AMLS method is a domain decomposition method for the solution of elliptic PDE eigenvalue problems which has been shown to be very efficient especially when a large number of eigenpairs is sought. In general the AMLS method is only applicable to these kind of continuous eigenvalue problems where the corresponding discretisation leads to an algebraic eigenvalue problem of the form $Kx = \lambda Mx$ where $K, M \in \mathbb{R}^{N\times N}$ are symmetric sparse matrices. However, the discretisation of an integral equation eigenvalue problem leads to a discrete problem where the matrix K is typically dense, since a non-local integral operator is involved in the equation. In the new method, which is introduced in this paper, the domain decomposition technique of classical AMLS is generalised to eigenvalue problems $Kx = \lambda Mx$ where K, M are symmetric and possibly dense, and which is therefor applicable for the solution of integral equation eigenvalue problems. To make out of this an efficient eigensolver, the new domain decomposition method is combined with a recursive approach and with the concept of hierarchical matrices.

Mathematics Subject Classification (2000) 45B05, 45C05, 65F15, 65F30, 65R20
Keywords Automated multi-level substructuring, hierarchical matrices, integral equation eigenvalue problem, domain decomposition method, dense eigenvalue problem

1 Introduction

Integral equation eigenvalue problems are typically solved using the finite element discretisation where the arising discrete eigenvalue problem is solved by an iterative algebraic eigensolver or a direct method (e.g., QR-algorithm). Domain decomposition techniques, which are available for example for the solution of elliptic PDE eigenvalue problems (see, e.g., [7, 30]), however, are not known (to the best of the author's knowledge) for integral equation eigenvalue problems. In this paper the very first domain decomposition method for the solution of integral equation eigenvalue problems is introduced. This new domain decomposition method is motivated by the so-called *automated multi-level substructuring*

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

(short AMLS) method. The AMLS method was mainly developed by Bennighof and co-authors [5, 7, 23] for the solution of elliptic PDE eigenvalue problems, and it has been shown that AMLS is very efficient especially when a large number of eigenpairs is sought [6, 23, 26]. In general the AMLS method is only applicable to these kind of continuous eigenvalue problems where the corresponding discretisation leads to an algebraic eigenvalue problem of the form $Kx = \lambda Mx$ where the matrices $K, M \in \mathbb{R}^{N\times N}$ are symmetric sparse. However, the discretisation of integral equation eigenvalue problems leads to a discretised problem $Kx = \lambda Mx$ where the matrix K is typically dense, because a non-local integral operator is involved in the equation. In the new method, which is introduced in the following and which is called *dense AMLS method*, the domain decomposition technique of classical AMLS is generalised to eigenvalue problems $Kx = \lambda Mx$ where K, M are symmetric and possibly dense, and which is therefore applicable for the solution of integral equation eigenvalue problems. To improve the efficiency of this new domain decomposition technique, the dense AMLS method is combined with a recursive approach and with the concept of the *hierarchical matrices* (short $\mathcal{H}$-matrices).

The remainder of this paper is organised as follows: In Section 2 the integral equation eigenvalue problem and the underlying problem setting is introduced. After this, in Section 3, the basic steps of the classical AMLS method are summarised. In Section 4 the new dense AMLS method is introduced and in Section 5 numerical results are presented. Finally, in Section 6 a recursive version of dense AMLS is introduced which is combined with the concept of $\mathcal{H}$-matrices. A summary and conclusion of this work is given in Section 7.

2 Problem Description

In this paper we consider the continuous eigenvalue problem

$$(A\,u)(x) = \lambda u(x) \quad \text{for all } x \in \Omega \tag{1}$$

where $\Omega \subset \mathbb{R}^d$ is an open bounded domain ($d \in \mathbb{N}$), A is a linear Fredholm integral operator of the form

$$(A\,u)(x) := \int_\Omega k(x,y)u(y)\,\mathrm{d}y \quad \text{with } x \in \Omega \tag{2}$$

with kernel function $k : \Omega \times \Omega \to \mathbb{R}$, and where in (1) a suitable eigenfunction $u : \Omega \to \mathbb{C}$ and the associated eigenvalue $\lambda \in \mathbb{C}$ are sought. Throughout the paper the following definitions are used for the kernel function k:

- k is called *continuous* iff $k \in C^0(\overline{\Omega} \times \overline{\Omega})$.
- k is called *weak singular* iff $k \in C^0(\{(x,y) \in \overline{\Omega} \times \overline{\Omega} : x \neq y\})$ and there exist constants $C > 0$ and $\alpha \in [0, d)$ such that

$$|k(x,y)| \leq C|x-y|^{-\alpha} \qquad \text{for all } x, y \in \overline{\Omega} \text{ with } x \neq y.$$

- k is called *symmetric* iff $k(x,y) = k(y,x)$ for all $x, y \in \overline{\Omega}$ with $x \neq y$.

From the theory of linear integral equations (see, e.g., [20]) the following result is known:

Theorem 2.1 (Existence of Eigensolutions) *Let the kernel function k in (2) be continuous or weak singular with $\alpha \in [0, d/2)$, let k be symmetric and assume that $k \neq 0$. Then the integral operator $A : L^2(\Omega) \to L^2(\Omega)$ in (2) is selfadjoint and compact, and hence eigenvalue problem (1) possesses a countable family of eigensolutions*

$$\big(\lambda_j, u_j\big)_{j=1}^{\infty} \in \mathbb{R} \times L^2(\Omega) \setminus \{0\} \tag{3}$$

where all eigenvalues λ_j are ordered with respect to their multiplicity[1] and absolute value such that $|\lambda_1| \geq |\lambda_2| \geq \ldots \geq 0$. In particular, it holds

i) All eigenvalues λ_j are real and we have $\lambda_j \xrightarrow{j\to\infty} 0$.

ii) The eigenspace $E(\lambda_j) \subset L^2(\Omega)$ of the eigenvalue λ_j, which is defined by

$$E(\lambda_j) := \operatorname{span}\Big\{ u \in L^2(\Omega) \ : \ A\,u = \lambda_j u \Big\}, \tag{4}$$

is finite-dimensional for $\lambda_j \neq 0$.

iii) If it holds $\lambda_j \neq \lambda_k$ then the corresponding eigenfunctions u_j and u_k are orthogonal with respect to the $L^2(\Omega)$-inner product, i.e., it holds $(u_j, u_k)_0 := \int_\Omega u_j(x) u_k(x)\,\mathrm{d}x = 0$.

iv) The eigenfunctions $\big(u_j\big)_{j=1}^{\infty}$ form a basis of the Hilbert space $L^2(\Omega)$ and without loss of generality it can be assumed that all eigenfunctions are orthonormal with respect to $(\cdot,\cdot)_0$.

Multiplying (1) by $v \in L^2(\Omega)$ and integrating over Ω we obtain the equivalent variational eigenvalue problem

$$\begin{cases} \text{find } \ (\lambda, u) \in \mathbb{R} \times L^2(\Omega) \setminus \{0\} \ \text{ such that} \\ a(u,v) \ = \ \lambda\,(u,v)_0 \qquad \forall\, v \in L^2(\Omega) \end{cases} \tag{5}$$

with the symmetric bilinear form $a(u,v) := \int_\Omega \int_\Omega v(x) k(x,y) u(y)\,\mathrm{d}y\,\mathrm{d}x$. We approximate solutions of the continuous eigenvalue problem (1) and (5) by discretisation: Using a conforming finite element space $V_h \subset L^2(\Omega)$ with dimension N_h and nodal basis $\big(\varphi_i^{(h)}\big)_{i=1}^{N_h}$ the eigenvalue problem (5) is discretised by

$$\begin{cases} \text{find } (\lambda^{(h)}, x^{(h)}) \in \mathbb{R} \times \mathbb{R}^{N_h} \setminus \{0\} \text{ with} \\ K^{(h)}\, x^{(h)} \ = \ \lambda^{(h)}\, M^{(h)}\, x^{(h)} \end{cases} \tag{6}$$

where the stiffness and mass matrix

$$K^{(h)} := \Big(a(\varphi_j^{(h)}, \varphi_i^{(h)}) \Big)_{i,j=1}^{N_h} \in \mathbb{R}^{N_h \times N_h} \quad \text{and} \quad M^{(h)} := \Big((\varphi_j^{(h)}, \varphi_i^{(h)})_0 \Big)_{i,j=1}^{N_h} \in \mathbb{R}^{N_h \times N_h} \tag{7}$$

are both symmetric. In contrast to the sparse and positive definite mass matrix M, the stiffness matrix K is in general dense (i.e., nearly all entries of K are nonzero) and in general not positive definite.

From the approximation theory of integral equation eigenvalue problems follows that the discrete eigensolutions $(\lambda^{(h)}, u^{(h)}) := (\lambda^{(h)}, \mathcal{P} x^{(h)})$ are approximating the continuous eigensolutions (λ, u) of (1), where $u^{(h)}$ is associated to the discrete problem (6) via

$$\mathcal{P} : \mathbb{R}^{N_h} \to V_h \subset L^2(\Omega) \quad \text{with} \quad x^{(h)} \mapsto \sum_{i=1}^{N_h} x_i^{(h)}\, \varphi_i^{(h)}\,.$$

For the precise formulation of the corresponding approximation result an additional notation of the continuous and discrete eigensolutions is introduced which distinguishes between positive and negative eigensolutions: In the following eigensolutions of the continuous and discrete problem associated to the positive eigenvalues are denoted by

$$\begin{aligned} \big(\overset{+}{\lambda}_j, \overset{+}{u}_j\big) \in \mathbb{R} \times L^2(\Omega) \setminus \{0\} \qquad & \text{where} \quad \overset{+}{\lambda}_1 \geq \overset{+}{\lambda}_2 \geq \ldots \geq 0, \\ \big(\overset{+}{\lambda}_j^{(h)}, \overset{+}{u}_j^{(h)}\big) \in \mathbb{R} \times V_h \setminus \{0\} \qquad & \text{where} \quad \overset{+}{\lambda}_1^{(h)} \geq \overset{+}{\lambda}_2^{(h)} \geq \ldots \geq 0 \end{aligned}$$

[1] This means that the eigenvalues in (3) are repeated according to the dimension of the corresponding eigenspace

and eigensolutions associated to the negative eigenvalues by

$$\begin{aligned} (\bar{\lambda}_j, \bar{u}_j) &\in \mathbb{R} \times L^2(\Omega) \setminus \{0\} \qquad \text{where} \quad \bar{\lambda}_1 \leq \bar{\lambda}_2 \leq \ldots \leq 0, \\ (\bar{\lambda}_j^{(h)}, \bar{u}_j^{(h)}) &\in \mathbb{R} \times V_h \setminus \{0\} \qquad \text{where} \quad \bar{\lambda}_1^{(h)} \leq \bar{\lambda}_2^{(h)} \leq \ldots \leq 0. \end{aligned}$$

Theorem 2.2 (Qualitative Convergency Results) *Let the assumptions of Theorem 2.1 be valid. Consider eigenvalue problem* (1) *and its finite element discretisation* (6) *where the eigensolutions are indexed as above, and where the finite element space $V_h \subset L^2(\Omega)$ fulfils approximation property*

$$\lim_{h \to 0} \inf_{v^{(h)} \in V_h} \|u - v^{(h)}\|_0 = 0 \qquad \text{for all } u \in L^2(\Omega) \tag{8}$$

with $\|\cdot\|_0 := (\cdot,\cdot)_0^{1/2}$. Then the discrete eigenvalues are approximating the continuous ones, i.e., it holds

$$\overset{+}{\lambda}_j^{(h)} \xrightarrow{h \to 0} \overset{+}{\lambda}_j \quad \text{and} \quad \bar{\lambda}_j^{(h)} \xrightarrow{h \to 0} \bar{\lambda}_j \qquad \text{for } j \in \mathbb{N};$$

and for the associated discrete eigenfunctions $\overset{+}{u}_j^{(h)}$ and $\bar{u}_j^{(h)}$ [assuming that $\|\overset{+}{u}_j^{(h)}\|_0 = 1$ and $\|\bar{u}_j^{(h)}\|_0 = 1$] exist subsequences which converge in $L^2(\Omega)$ to an eigenfunction $\overset{+}{u} \in E(\overset{+}{\lambda}_j)$ and to an eigenfunction $\bar{u} \in E(\bar{\lambda}_j)$ when $h \to 0$.

Proof: The result for the convergence of the eigenvalues follows directly from the well-known Minimum-Maximum principle

$$\overset{+}{\lambda}_j = \max_{\substack{H_j \subset L^2(\Omega), \\ \dim H_j = j}} \min_{u \in H_j \setminus \{0\}} \frac{a(u,u)}{(u,u)_0} \quad \text{and} \quad \overset{+}{\lambda}_j^{(h)} = \max_{\substack{H_j \subset V_h, \\ \dim H_j = j}} \min_{u \in H_j \setminus \{0\}} \frac{a(u,u)}{(u,u)_0} \quad \text{for } j = 1, 2, \ldots,$$

$$\bar{\lambda}_j = \min_{\substack{H_j \subset L^2(\Omega), \\ \dim H_j = j}} \max_{u \in H_j \setminus \{0\}} \frac{a(u,u)}{(u,u)_0} \quad \text{and} \quad \bar{\lambda}_j^{(h)} = \min_{\substack{H_j \subset V_h, \\ \dim H_j = j}} \max_{u \in H_j \setminus \{0\}} \frac{a(u,u)}{(u,u)_0} \quad \text{for } j = 1, 2, \ldots$$

related to compact selfadjoint operators, from the continuity of the Rayleigh quotient $R(u) := a(u,u)/(u,u)_0$, and from the approximation property (8) of V_h. The result for the convergence of the eigenfunctions is a combination of [19, Satz 4.8.15] and the convergence result of the eigenvalue approximation described above. ■

When eigensolutions (λ_j, u_j) of integral equation eigenvalue problems are approximated by the finite element method the following point has to be noted: Only the by magnitude largest eigenvalues λ_j and their corresponding eigenfunctions u_j can be well approximated by the finite element space V_h because the approximation error increases with by magnitude decreasing eigenvalue (see, e.g., numerical results from Section 5).

Hence, in the following we are interested only in computing a portion of the eigenpairs of the discrete problem (6), e.g., the by magnitude largest

$$n_{\text{es}} = C N_h^{1/3} \in \mathbb{N} \quad \text{or} \quad n_{\text{es}} = C N_h^{1/2} \in \mathbb{N} \tag{9}$$

eigenpairs (with some $C > 0$).

Depending on the number of sought eigenpairs, different approaches are better suited for the solution of the algebraic eigenvalue problem (6): If the number of sought eigenpairs n_{es} is rather small, e.g., if $n_{\text{es}} = 5$, an iterative algebraic eigensolver such as the Lanczos method [2, 18] is a good choice for the solution of (6). If the number of sought eigenpairs approaches N_h, it is advisable to use instead a cubic scaling direct method such as the QR algorithm [31].

However, since we are interested in a large number of eigenpairs where n_{es} is, e.g., of the size (9), neither an iterative eigensolver nor a direct method might be a good choice for the solution of the dense problem (6). Using domain decomposition techniques might be a new approach for the efficient solution of (6). The domain decomposition techniques, which is applied in the AMLS method for the solution of elliptic PDE eigenvalue value problems, has been shown to be very efficient, especially when the number of sought eigenpairs is, e.g., of the size (9) [see, e.g., [13, 14]]. The efficiency of the AMLS method for the solution of elliptic PDE eigenvalue problems motivates to transfer the domain decomposition technique of AMLS to integral equation eigenvalue problems. For doing this the domain decomposition technique of classical AMLS is briefly described in the following section.

3 The Classical AMLS Method for Sparse Eigenvalue Problems

This section summarises the basic steps of the classical AMLS method for the solution of an elliptic PDE eigenvalue problem. The following discussion is restricted to a purely algebraic setting and only to a single-level version of the AMLS method. To see a multi-level version of AMLS and to see how AMLS is motivated in the context of a continuous setting, it is referred, e.g., to [7, 14].

The initial point of this section is the elliptic PDE eigenvalue problem

$$\begin{cases} Lu = \lambda u & \text{in } \Omega, \\ u = 0 & \text{on } \partial\Omega \end{cases} \tag{10}$$

where $\Omega \subset \mathbb{R}^d$ is a bounded Lipschitz domain and L is a uniformly elliptic second order partial differential operator in divergency form $Lu = -\text{div}\big(A\nabla u\big) + cu$ with $A \in (L^\infty(\Omega))^{d\times d}$, $c \in L^\infty(\Omega)$ and $c \geq 0$. We approximate solutions of the continuous eigenvalue problem (10) by discretisation. Using a conforming N-dimensional finite element space V_h the eigenvalue problem (10), respectively its weak formulation, is discretised by

$$\begin{cases} \text{find } (\lambda, x) \in \mathbb{R} \times \mathbb{R}^N \setminus \{0\} \text{ with} \\ K\,x \;=\; \lambda\, M\, x \end{cases} \tag{11}$$

where the stiffness matrix $K \in \mathbb{R}^{N\times N}$ and the mass matrix $M \in \mathbb{R}^{N\times N}$ are sparse and symmetric positive definite, and where the eigenpairs

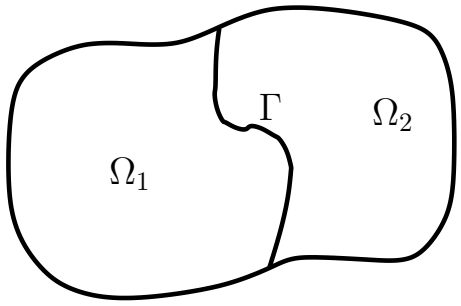
$$\big(\lambda_j, x_j\big)_{j=1}^{N} \in \mathbb{R}_{>0} \times \mathbb{R}^N \setminus \{0\} \qquad \text{with} \quad \lambda_j \leq \lambda_{j+1} \tag{12}$$

of problem (11) have all positive eigenvalues. For reasons of convenience the upper index of $\lambda^{(h)}, x^{(h)}, K^{(h)}$ and $M^{(h)}$, which is indicating the mesh width h of the underlying finite element discretisation, is left out in (11).

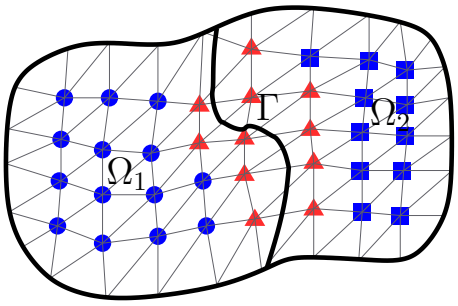
The discrete eigenpairs (λ_j, x_j) of problem (11) provide approximations of the eigensolutions of problem (10). However, it has to be noted that only the discrete eigenpairs associated to the smallest eigenvalues provide good approximations of continuous eigensolutions since in the context of elliptic PDE eigenvalue problems the error of the finite element approximation increases with the size of the eigenvalue (see, e.g., [3, 29]). Hence, we are interested only in the eigenpairs (12) which are associated to the smallest eigenvalues, where the number of sought eigenpairs n_{es} is, e.g., of the size (9).

To solve eigenvalue problem (11) by the AMLS method, the underlying domain Ω in (10) is substructured into two non-overlapping subdomains Ω_1 and Ω_2 with

$$\overline{\Omega} = \overline{\Omega}_1 \cup \overline{\Omega}_2 \qquad \text{and} \qquad \Omega_1 \cap \Omega_2 = \emptyset,$$



(a) Substructuring of the domain Ω into the non-overlapping subdomains Ω_1 and Ω_2, where $\overline{\Omega} = \overline{\Omega}_1 \cup \overline{\Omega}_2$, $\Omega_1 \cap \Omega_2 = \emptyset$ and $\Gamma := \overline{\Omega}_1 \cap \overline{\Omega}_2$.



(b) DOF are associated to the interface Γ when their basis functions have supports that are intersecting the interface Γ.

Figure 1: Substructuring of Ω with triangulation. Degrees of freedom (DOF) of the finite element space of continuous and piecewise affine functions are indicated by circles if they are associated to Ω_1, by squares if associated to Ω_2, and by triangles if associated to Γ.

which share the interface $\Gamma := \overline{\Omega}_1 \cap \overline{\Omega}_2$ (an example is given in Figure 1a). Since the matrices $K, M \in \mathbb{R}^{N\times N}$ in (11) result from a finite element discretisation each row and column index is associated with a basis function that has typically a small support. Using the substructuring of Ω the row and column indices of K and M are reordered in such a way that it holds

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

where $K_{ij}, M_{ij} \in \mathbb{R}^{N_i \times N_j}$ and $N_1 + N_2 + N_3 = N$. The labels $\Omega_1, \Omega_2, \Gamma$ used in (13) are indicating to which subset the row and column indices are associated, i.e., if the supports of the corresponding basis functions are inside Ω_i or if they are intersecting Γ (cf. Figure 1b).

In the next step of AMLS a block $\mathrm{LDL}^{\mathrm{T}}$-decomposition is performed in order to block diagonalise the stiffness matrix K by $K = L\widetilde{K}L^T$ where

$$L := \begin{bmatrix} \mathrm{Id} & & \\ & \mathrm{Id} & \\ K_{31}K_{11}^{-1} & K_{32}K_{22}^{-1} & \mathrm{Id} \end{bmatrix} \in \mathbb{R}^{N\times N} \quad \text{and} \quad \widetilde{K} = \mathrm{diag}\big[K_{11}, K_{22}, \widetilde{K}_{33}\big], \tag{14}$$

and where 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 $\mathrm{diag}[K_1, K_2]$ in K. The mass matrix M is transformed correspondingly by computing $\widetilde{M} := L^{-1}ML^{-T}$ with

$$\widetilde{M} = \begin{bmatrix} M_{11} & & \widetilde{M}_{13} \\ & M_{22} & \widetilde{M}_{23} \\ \widetilde{M}_{31} & \widetilde{M}_{32} & \widetilde{M}_{33} \end{bmatrix}, \tag{15}$$

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

$$\widetilde{M}_{3i} \;=\; M_{3i} - K_{3i}K_{ii}^{-1}M_{ii} \qquad \text{and} \qquad \widetilde{M}_{i3} \;=\; \widetilde{M}_{3i}^T \qquad (\text{for } i = 1, 2),$$
$$\widetilde{M}_{33} \;=\; M_{33} - \sum_{i=1}^{2} \big(K_{3i}K_{ii}^{-1}M_{i3} + M_{3i}K_{ii}^{-1}K_{i3} - K_{3i}K_{ii}^{-1}M_{ii}K_{ii}^{-1}K_{i3} \big).$$

Note that the eigenvalue problems $(\widetilde{K}, \widetilde{M})$ and (K, M) are equivalent, i.e., the eigenvalues of both problems are equal and if $\widetilde{x}$ is an eigenvector of problem $(\widetilde{K}, \widetilde{M})$ then $x = L^{-T}\widetilde{x}$ is an eigenvector of (K, M). The reason for the performed problem transformation from (K, M) to $(\widetilde{K}, \widetilde{M})$ is explained in the best way by describing AMLS in a continuous setting, however, this issue is not discussed in this paper and instead it is referred to, e.g., [7, 13, 28].

In the next step of AMLS partial eigensolutions of the subproblems (K_{11}, M_{11}), (K_{22}, M_{22}) and $(\widetilde{K}_{33}, \widetilde{M}_{33})$ are computed. To be more precise, only those eigenpairs of the subproblems are computed which belong to the smallest $k_i \in \mathbb{N}$ eigenvalues for given $k_i \leq N_i$ $(i = 1, 2, 3)$. In the following these partial eigensolutions are

$$K_{ii}\,\widetilde{S}_i \;=\; M_{ii}\,\widetilde{S}_i\,\widetilde{D}_i \quad \text{for } i = 1, 2 \qquad \text{and} \qquad \widetilde{K}_{33}\,\widetilde{S}_3 \;=\; \widetilde{M}_{33}\,\widetilde{S}_3\,\widetilde{D}_3, \tag{16}$$

where the diagonal matrix $\widetilde{D}_i \in \mathbb{R}^{k_i \times k_i}$ contains the k_i smallest eigenvalues, and where the matrix $\widetilde{S}_i \in \mathbb{R}^{N_i \times k_i}$ contains column-wise the associated eigenvectors $(i = 1, 2, 3)$ which are normalised by $\widetilde{S}_i^T M_{ii} \widetilde{S}_i = \mathrm{Id}$ $(i = 1, 2)$ and $\widetilde{S}_3^T \widetilde{M}_{33} \widetilde{S}_3 = \mathrm{Id}$.

It is important to note that the original eigenvalue problem (K, M) is not solved just by computing the eigenpairs of the subproblems (K_{11}, M_{11}), (K_{22}, M_{22}) and $(\widetilde{K}_{33}, \widetilde{M}_{33})$. However, it could be shown (see, e.g., [7, 9, 10]) that the eigenvectors of the subproblems, which are associated to the smallest eigenvalues, are very well suited to form a subspace for the approximation of the original eigenvalue problem (K, M). In order to approximate eigenvalue problem (K, M), respectively the sought eigenpairs of (K, M), in the next step of AMLS the block diagonal matrix

$$Z := \mathrm{diag}\left[\widetilde{S}_1, \widetilde{S}_2, \widetilde{S}_3\right] \in \mathbb{R}^{N \times \bar{k}} \qquad \text{with } \bar{k} := k_1 + k_2 + k_3 \ll N$$

is defined, and the *reduced matrices* $\widehat{K} := Z^T\,\widetilde{K}\,Z$ and $\widehat{M} := Z^T\,\widetilde{M}\,Z$ are computed where it holds

$$\widehat{K} = \mathrm{diag}\left[\widetilde{D}_1, \widetilde{D}_2, \widetilde{D}_3\right] \in \mathbb{R}^{\bar{k} \times \bar{k}} \qquad \text{and} \qquad \widehat{M} = \begin{bmatrix} \mathrm{Id} & & \widehat{M}_{13} \\ & \mathrm{Id} & \widehat{M}_{23} \\ \widehat{M}_{31} & \widehat{M}_{32} & \mathrm{Id} \end{bmatrix} \in \mathbb{R}^{\bar{k} \times \bar{k}}.$$

With the reduced matrices a *reduced eigenvalue problem*

$$\begin{cases} \text{find } (\widehat{\lambda}, \widehat{x}\,) \in \mathbb{R} \times \mathbb{R}^{\bar{k}} \setminus \{0\} \text{ with} \\ \widehat{K}\,\widehat{x} \;=\; \widehat{\lambda}\,\widehat{M}\,\widehat{x} \end{cases} \tag{17}$$

arises which possesses the eigenpairs

$$\big(\widehat{\lambda}_j, \widehat{x}_j\big)_{j=1}^{\bar{k}} \in \mathbb{R}_{>0} \times \mathbb{R}^{\bar{k}} \setminus \{0\} \qquad \text{and} \quad \widehat{\lambda}_j \leq \widehat{\lambda}_{j+1}.$$

Finally, in the last steps of AMLS the smallest n_{es} eigenpairs of (17) are computed and the eigenvectors $\widehat{x}_j$ of the reduced problem are transformed by computing

$$\widehat{y}_j := L^{-T} Z\,\widehat{x}_j \qquad \text{with } j = 1, \ldots, n_{\mathrm{es}}. \tag{18}$$

The vectors $\widehat{y}_j$ are Ritz-vectors of the original eigenvalue problem (K, M) respective to the subspace spanned by the columns of the matrix $L^{-T}Z$, and $\widehat{\lambda}_j$ are the respective Ritz-values. The Ritz-pairs

$$\left(\widehat{\lambda}_j, \widehat{y}_j\right)_{j=1}^{n_{\text{es}}} \in \mathbb{R}_{>0} \times \mathbb{R}^N \setminus \{0\} \qquad \text{with} \quad \widehat{\lambda}_j \leq \widehat{\lambda}_{j+1} \tag{19}$$

are approximating the sought smallest n_{es} eigenpairs of the eigenvalue problem (K, M).

Remark 3.1 *i) Note that the Ritz-pairs $(\widehat{\lambda}_j, \widehat{y}_j)$, which are computed by AMLS, are primarily used to approximate the eigensolutions of the continuous problem* (10) *and not the eigenpairs of the discretised problem (K, M). Correspondingly the approximation error of AMLS is influenced by both the modal truncation performed in* (16) *and the finite element discretisation. In contrast to the AMLS method, the approximation error of a classical approach, like the Lanczos method [2] or the QR-algorithm [31], is only influenced by the finite element discretisation since (almost) exact eigenpairs of the discrete problem (K, M) are computed. This means as long as the error caused by the modal truncation in AMLS is of the same order as the discretisation error, the eigenpair approximations computed by AMLS are of comparable quality as the eigenpair approximations computed by a classical approach.*

ii) How many eigenvectors have to be computed in (16) *for each subproblem is not easy to answer. On the one hand k_i should be large enough in order to keep enough spectral information from each subproblem so that sufficiently good eigenpair approximations are obtained from the reduced problem $(\widehat{K}, \widehat{M})$ [e.g., AMLS computes exact eigenpairs of the discrete problem (K, M) if $k_i = N_i$ for $i = 1, 2, 3$]. On the other hand k_i should be small in order to obtain a reduced problem $(\widehat{K}, \widehat{M})$ of small size which can be easily solved. Several heuristic approaches have been discussed in literature for the eigenpair selection in* (16)*: One possible strategy is to select from each subproblem only those eigenvectors whose eigenvalues are smaller than a given* truncation bound $\omega > 0$ *(see, e.g., [11, 32]). Another possible strategy, which is motivated by approximation properties of the finite element discretisation of elliptic PDE eigenvalue problems, is to compute only these eigenpairs of the subproblems which belong, for example, to the smallest k_i eigenvalues with*

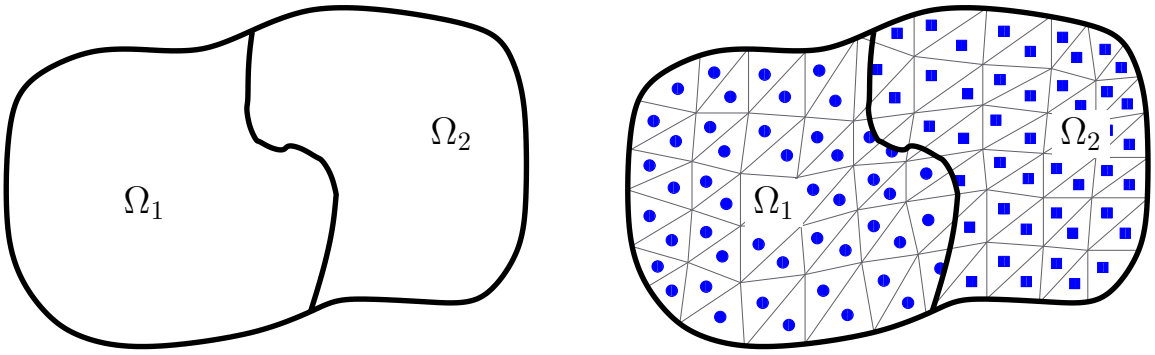
$$k_i = CN_i^{1/3} \in \mathbb{N} \quad \text{or} \quad k_i = CN_i^{1/2} \in \mathbb{N} \tag{20}$$

and some constant $C > 0$ (see [13, 14] for details).

iii) Since the number k_i of selected eigenpairs in (16) *is typically quite small, the size $\bar{k}$ of the reduced eigenvalue problem $(\widehat{K}, \widehat{M})$ is much smaller than the size N of the original problem (K, M), and hence the reduced problem is typically much easier to solve than the original problem. If for example the mode selection described in ii) is used with $k_i = CN_i^{1/3}$ 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 $\mathcal{O}(N)$.*

4 Introduction of the new dense AMLS method

The new domain decomposition method for dense eigenvalue problems, which is introduced in this section and which is called in the following *dense AMLS method*, is motivated by the classical AMLS method from the previous section. The initial point of dense AMLS is the finite element discretisation (6) of the continuous integral equation eigenvalue problem (1). For reasons of convenience the upper index of $\lambda^{(h)}, x^{(h)}, K^{(h)}$ and $M^{(h)}$, which is indicating in (6) the mesh width h of the underlying



(a) Substructuring of the domain Ω into the non-overlapping subdomains Ω_1 and Ω_2, where $\overline{\Omega} = \overline{\Omega}_1 \cup \overline{\Omega}_2$ and $\Omega_1 \cap \Omega_2 = \emptyset$.

(b) DOF are associated to Ω_i when the nodal points of the corresponding basis functions are elements of Ω_i.

Figure 2: Substructuring of Ω with triangulation. Degrees of freedom (DOF) of the finite element space of piecewise constant basis functions are indicated by circles if they are associated to Ω_1 and by squares if associated to Ω_2.

finite element discretisation, is left out in this particular section, i.e., in the following we consider the eigenvalue problem

$$\begin{cases} \text{find } (\lambda, x) \in \mathbb{R} \times \mathbb{R}^N \setminus \{0\} \text{ with} \\ K\,x \;=\; \lambda\, M\,x \end{cases} \tag{21}$$

with the eigenpairs

$$\left(\lambda_j, x_j\right)_{j=1}^{N} \in \mathbb{R} \times \mathbb{R}^N \setminus \{0\} \qquad \text{where} \quad |\lambda_j| \geq |\lambda_{j+1}|$$

and with $N := N_h = \dim V_h$. To avoid misunderstandings, it is explicitly noted that in this section λ and λ_j are interpreted as the eigenvalues of the discrete problem (21) and not as the eigenvalues of the continuous problem (1). Furthermore, we are interested only in computing the by magnitude largest eigenpairs of (21), where the number of sought eigenpairs n_{es} is, e.g., of the size (9).

In the first step of dense AMLS the domain Ω is subdivided into two non-overlapping subdomains Ω_1 and Ω_2 (cf. Figure 2a) such that

$$\overline{\Omega} = \overline{\Omega}_1 \cup \overline{\Omega}_2 \qquad \text{and} \qquad \Omega_1 \cap \Omega_2 = \emptyset. \tag{22}$$

Since the matrices $K, M \in \mathbb{R}^{N \times N}$ in (21) result from a finite element discretisation each row and column index is associated with a basis function, which has typically a small support and which has some nodal point (this might be, e.g., the midpoint of the corresponding finite element). Using the substructuring of Ω the row and column indices of K and M are reordered in such a way that

$$K \;=\; \begin{matrix} \\ \Omega_1 \\ \Omega_2 \end{matrix} \begin{matrix} \begin{matrix} \Omega_1 & \Omega_2 \end{matrix} \\ \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \end{matrix} \qquad \text{and} \qquad M \;=\; \begin{matrix} \\ \Omega_1 \\ \Omega_2 \end{matrix} \begin{matrix} \begin{matrix} \Omega_1 & \Omega_2 \end{matrix} \\ \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \end{matrix} \tag{23}$$

holds with $K_{ij}, M_{ij} \in \mathbb{R}^{N_i \times N_j}$ and $N_1 + N_2 = N$. The labels Ω_1 and Ω_2 in (23) are indicating to which subset the row and column indices are associated. In contrast to the matrix partitioning performed in classical AMLS (which depends on the supports of the basis functions), the matrix partitioning in dense AMLS depends only on the nodal points of the corresponding basis functions. A row or column

index is associated to Ω_1 if the nodal point of the corresponding basis function is inside Ω_1, otherwise the index is associated to Ω_2 (cf. Figure 2b).

In the next step of dense AMLS a block LDL$^{\mathrm{T}}$-decomposition is performed in order to block diagonalise the matrix K by $K = L\widetilde{K}L^T$ where

$$L := \begin{bmatrix} \mathrm{Id} & \\ K_{21}K_{11}^{-1} & \mathrm{Id} \end{bmatrix} \in \mathbb{R}^{N\times N} \qquad \text{and} \qquad \widetilde{K} = \mathrm{diag}\big[K_{11}, \widetilde{K}_{22}\big]. \tag{24}$$

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

$$\widetilde{K}_{22} = K_{22} - K_{21}K_{11}^{-1}K_{12}$$

is the *Schur complement* of K_{11} in K. The matrix M is transformed correspondingly by computing $\widetilde{M} := L^{-1}ML^{-T}$ with

$$\widetilde{M} = \begin{bmatrix} M_{11} & \widetilde{M}_{12} \\ \widetilde{M}_{21} & \widetilde{M}_{22} \end{bmatrix} \tag{25}$$

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

$$\begin{aligned} \widetilde{M}_{21} &= M_{21} - K_{21}K_{11}^{-1}M_{11} \qquad \text{and} \qquad \widetilde{M}_{12} = \widetilde{M}_{21}^T, \\ \widetilde{M}_{22} &= M_{22} - K_{21}K_{11}^{-1}M_{12} - M_{21}K_{11}^{-1}K_{12} - K_{21}K_{11}^{-1}M_{11}K_{11}^{-1}K_{12}. \end{aligned}$$

The eigenvalue problems $(\widetilde{K}, \widetilde{M})$ and (K, 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).

In the next step of dense AMLS partial eigensolutions of the subproblems (K_{11}, M_{11}) and $(\widetilde{K}_{22}, \widetilde{M}_{22})$ are computed. However, in contrast to the classical AMLS method, only those eigenpairs of the subproblems are computed which belong to the by magnitude largest $k_i \in \mathbb{N}$ eigenvalues for given $k_i \le N_i$ $(i = 1, 2)$. In the following these partial eigensolutions are

$$K_{11}\,\widetilde{S}_1 = M_{11}\,\widetilde{S}_1\,\widetilde{D}_1 \qquad \text{and} \qquad \widetilde{K}_{22}\,\widetilde{S}_2 = \widetilde{M}_{22}\,\widetilde{S}_2\,\widetilde{D}_2, \tag{26}$$

where the diagonal matrix $\widetilde{D}_i \in \mathbb{R}^{k_i\times k_i}$ contains the k_i selected eigenvalues, and the matrix $\widetilde{S}_i \in \mathbb{R}^{N_i\times k_i}$ contains column-wise the associated eigenvectors $(i = 1, 2)$ which are normalised by $\widetilde{S}_1^T M_{11}\widetilde{S}_1 = \mathrm{Id}$ and $\widetilde{S}_2^T\widetilde{M}_{22}\widetilde{S}_2 = \mathrm{Id}$.

Remark 4.1 (Mode Selection) *Motivated by the mode selection strategy described in Remark 3.1 ii), it is proposed to compute in the dense AMLS method (which has been introduced in the framework of integral equation eigenvalue problems) only these eigenpairs of* (26) *which belong to the by magnitude largest k_i eigenvalues and where k_i is chosen, for example, as in* (20).

In the next step of dense AMLS we define the block diagonal matrix

$$Z := \mathrm{diag}\left[\widetilde{S}_1, \widetilde{S}_2\right] \in \mathbb{R}^{N\times\bar{k}} \qquad \text{with } \bar{k} := k_1 + k_2 \ll N.$$

The $\bar{k}$-dimensional subspace which is spanned by the columns of the matrix $L^{-T}Z$ is potentially well suited to approximate the sought eigenpairs of (K, M). To approximate these eigenpairs, in the next step of dense AMLS the *reduced matrices* $\widehat{K} := Z^T\,\widetilde{K}\,Z$ and $\widehat{M} := Z^T\,\widetilde{M}\,Z$ are computed where it holds

$$\widehat{K} = \mathrm{diag}\left[\widetilde{D}_1, \widetilde{D}_2\right] \in \mathbb{R}^{\bar{k}\times\bar{k}} \qquad \text{and} \qquad \widehat{M} = \begin{bmatrix} \mathrm{Id} & \widehat{M}_{12} \\ \widehat{M}_{21} & \mathrm{Id} \end{bmatrix} \in \mathbb{R}^{\bar{k}\times\bar{k}},$$

task		matrix operations of dense AMLS
($\mathbb{T}1$)	partition matrices K and M	apply geometric bisection reordering as done in (23)
($\mathbb{T}2$)	block diagonalise the matrix K	$K = L\,\widetilde{K}\,L^T$
($\mathbb{T}3$)	transform M	$\widetilde{M} = L^{-1}\,M\,L^{-T}$
($\mathbb{T}4$)	compute partial eigensolutions of the subproblems	$K_{11}\,\widetilde{S}_1 = M_{11}\,\widetilde{S}_1\,\widetilde{D}_1$ with $\widetilde{S}_1 \in \mathbb{R}^{N_1\times k_1}$, $\widetilde{D}_1 \in \mathbb{R}^{k_1\times k_1}$ $\widetilde{K}_{22}\,\widetilde{S}_2 = \widetilde{M}_{22}\,\widetilde{S}_2\,\widetilde{D}_2$ with $\widetilde{S}_2 \in \mathbb{R}^{N_2\times k_2}$, $\widetilde{D}_2 \in \mathbb{R}^{k_2\times k_2}$
($\mathbb{T}5$)	define subspace	$Z := \operatorname{diag}\left[\widetilde{S}_1, \widetilde{S}_2\right] \in \mathbb{R}^{N\times\bar{k}}$ with $\bar{k} := k_1 + k_2$
($\mathbb{T}6$)	compute the reduced matrices	$\widehat{K} := Z^T\,\widetilde{K}\,Z \in \mathbb{R}^{\bar{k}\times\bar{k}}$ and $\widehat{M} := Z^T\,\widetilde{M}\,Z \in \mathbb{R}^{\bar{k}\times\bar{k}}$
($\mathbb{T}7$)	solve reduced eigenvalue problem	$\widehat{K}\,\widehat{x}_j = \widehat{\lambda}_j\,\widehat{M}\,\widehat{x}_j$ for $j = 1,\ldots,n_{\text{es}}$
($\mathbb{T}8$)	transform eigenvectors	$\widehat{y}_j := L^{-T}Z\,\widehat{x}_j$ for $j = 1,\ldots,n_{\text{es}}$

Table 1: Overview of the new dense AMLS method to compute eigenpair approximations $(\,\widehat{\lambda}_j, \widehat{y}_j)$ for the problem (K, M).

and a *reduced eigenvalue problem*

$$\begin{cases} \text{find } (\,\widehat{\lambda}, \widehat{x}\,) \in \mathbb{R}\times\mathbb{R}^{\bar{k}}\setminus\{0\} \text{ with} \\ \widehat{K}\,\widehat{x} \;=\; \widehat{\lambda}\,\widehat{M}\,\widehat{x} \end{cases} \tag{27}$$

is obtained which possesses the eigenpairs

$$\big(\,\widehat{\lambda}_j, \widehat{x}_j\big)_{j=1}^{\bar{k}} \in \mathbb{R}\times\mathbb{R}^{\bar{k}}\setminus\{0\} \qquad \text{and} \quad |\widehat{\lambda}_j| \geq |\widehat{\lambda}_{j+1}|.$$

At the end of dense AMLS the by magnitude largest n_{es} eigenpairs of (27) are computed and the eigenvectors $\widehat{x}_j$ of the reduced problem are transformed via

$$\widehat{y}_j := L^{-T}Z\,\widehat{x}_j \qquad \text{with } j = 1,\ldots,n_{\text{es}}. \tag{28}$$

The vectors $\widehat{y}_j$ are the Ritz-vectors of the original eigenvalue problem (K, M) respective to the subspace spanned by the columns of the matrix $L^{-T}Z$, and $\widehat{\lambda}_j$ are the respective Ritz-values. The Ritz-pairs

$$\big(\,\widehat{\lambda}_j, \widehat{y}_j\big)_{j=1}^{n_{\text{es}}} \in \mathbb{R}\times\mathbb{R}^{N}\setminus\{0\} \qquad \text{with} \quad |\widehat{\lambda}_j| \geq |\widehat{\lambda}_{j+1}| \tag{29}$$

are approximating the sought by magnitude largest n_{es} eigenpairs of the problem (K, M).

To summarise dense AMLS an overview of all necessary operations is given in Table 1 where the different tasks of the method are denoted by ($\mathbb{T}1$)–($\mathbb{T}8$).

Remark 4.2 (Rayleigh-Ritz Projection) *i) The dense AMLS method can be summarised also as follows: At first the subspace spanned by the columns of the matrix $Q := L^{-T}Z$ is computed [task ($\mathbb{T}1$) – ($\mathbb{T}4$)]. Depending on the mode selection strategy in task ($\mathbb{T}4$) this subspace is well suited to approximate the sought eigensolutions of (K, M). Thereafter, the Rayleigh-Ritz*

projection of problem (K, M) onto the subspace Q is computed, i.e., the reduced matrices $\widehat{K} = Q^T K Q$ and $\widehat{M} = Q^T K Q$ are computed [task ($\mathbb{T}$6)], the reduced eigenvalue problem $(\widehat{K}, \widehat{M})$ is solved [task ($\mathbb{T}$7)], where the pairs $(\widehat{\lambda}_j, Q\widehat{x}_j)$ [task ($\mathbb{T}$8)] finally provide approximations of the sought eigenpairs of (K, M).

ii) To solve eigenvalue problem (1) by dense AMLS the role of K and M in (21) can be interchanged so that problem $Mx = 1/\lambda Kx$ is considered instead. In this setting the eigenpairs of (M, K) associated to the by magnitude smallest eigenvalues are sought, and in task ($\mathbb{T}$4) the by magnitude smallest k_i eigenpairs have to be computed.

5 Numerical Results

The dense AMLS method has been implemented in C++ using the LAPACK/BLAS [1, 27] library. In the following we analyse numerically the dense AMLS method for the integral equation eigenvalue problem

$$\int_\Omega \log|x-y| u(y)\, \mathrm{d}y = \lambda u(x) \qquad \text{with } x \in \Omega := (0,1). \tag{30}$$

To solve problem (30) by the dense AMLS method or by a classical approach it is discretised by the finite element method using piecewise constant basis functions

$$\varphi_i^{(N)} := \begin{cases} 1 & \text{if } \frac{i-1}{N} \le x \le \frac{i}{N} \\ 0 & \text{otherwise} \end{cases} \qquad \text{for } i = 1, \ldots, N \tag{31}$$

where the interval $[0,1]$ is decomposed into N equispaced subintervals with mesh width $h := 1/N$. As described in (6) the discretisation results in the algebraic eigenvalue problem $(K^{(h)}, M^{(h)})$ whose discrete eigenpairs $(\lambda^{(h)}, x^{(h)})$ are approximating the sought eigensolutions (λ, u) of the continuous eigenvalue problem (30).

In the benchmarks presented in the following the eigenvalues $\lambda^{(h)}$ of the algebraic eigenvalue $(K^{(h)}, M^{(h)})$ have been all positive, and thus the discrete eigenvalues $\lambda_j^{(h)} := \overset{+}{\lambda}{}_j^{(h)}$ with $\lambda_1^{(h)} \ge \lambda_2^{(h)} \ge \ldots \ge 0$ are approximating the positive eigenvalues $\lambda_j := \overset{+}{\lambda}{}_j$ of (30) with $\lambda_1 \ge \lambda_2 \ge \ldots \ge 0$.

To evaluate the approximation accuracy of the eigenvalues $\widehat{\lambda}_j^{(h)}$ computed by dense AMLS (the upper index of $\widehat{\lambda}_j^{(h)}$ indicates the mesh width h of the underlying finite element discretisation), we compare the approximation quality of dense AMLS [cf. Remark 3.1 i)] with the approximation quality of a classical approach by examining the relative errors

$$\widehat{\delta}_j^{(h)} := \underbrace{\frac{|\,\lambda_j - \widehat{\lambda}_j^{(h)}\,|}{|\lambda_j|}}_{\substack{\text{relative error of} \\ \text{dense AMLS}}} \qquad \text{and} \qquad \delta_j^{(h)} := \underbrace{\frac{|\,\lambda_j - \lambda_j^{(h)}\,|}{|\lambda_j|}}_{\substack{\text{relative error} \\ \text{of discretisation}}} . \tag{32}$$

However, since for problem (30) the exact eigenvalues λ_j are not known, the eigenvalues λ_j in (32) are approximated by the discrete eigenvalues $\lambda_j^{(h_0)}$ which are associated to the very fine mesh width h_0:=2e-4 (h_0 corresponds to a discretisation with 5,000 DOF). Using these approximated relative errors the approximation quality of dense AMLS is benchmarked for problem (30) by investigating

$$\gamma_{n_{\mathrm{es}}}^{(h)} := \max\left\{ \widehat{\delta}_j^{(h)} \,/\, \delta_j^{(h)} \;:\; j = 1, \ldots, n_{\mathrm{es}} \right\} \tag{33}$$

j	$\widehat{\delta}_j^{(h)}$	$\delta_j^{(h)}$	$\widehat{\delta}_j^{(h)}/\delta_j^{(h)}$
1	1.93e-1	3.67e-6	5.25e+4
2	9.41e-2	2.74e-5	3.43e+3
3	7.72e-2	9.70e-5	7.95e+2
4	5.74e-2	2.02e-4	2.84e+2
5	5.10e-2	3.52e-4	1.45e+2
6	4.22e-2	5.38e-4	7.84e+1
7	4.02e-2	7.68e-4	5.23e+1
8	3.77e-2	1.03e-3	3.65e+1
9	3.67e-2	1.34e-3	2.73e+1
10	4.64e-2	1.68e-3	2.75e+1

Table 2: Relative errors of dense AMLS $\widehat{\delta}_j^{(h)}$ versus relative discretisation errors $\delta_j^{(h)}$ for the eigenvalue approximation.

which is the maximal ratio between the relative errors $\widehat{\delta}_j^{(h)}$ of dense AMLS and the relative discretisation errors $\delta_j^{(h)}$. If it holds

$$\gamma_{n_{\text{es}}}^{(h)} < 3 \tag{34}$$

it can be said that the approximation error of dense AMLS is of the same order as the discretisation error, and thus in this case the approximation quality of dense AMLS competes with the approximation quality of a classical approach [cf. Remark 3.1 i)].

The numerical results which are presented in the following have been obtained using the following setting:

- Eigenvalue problem (30) has been discretised with the mesh width h:=5e-3 leading to a discrete problem $(K^{(h)}, M^{(h)})$ with $N_h = 200$ DOF.
- The performed substructuring in task ($\mathbb{T}1$) of dense AMLS resulted in two subproblems with 100 DOF each. In task ($\mathbb{T}4$) in each subproblem the eigensolutions associated to the largest $k_i = 5$ eigenvalues have been computed leading to a reduced problem of size $\bar{k} = 10$. Hence, the dense AMLS method computed the eigenvalue approximations $\widehat{\lambda}_j^{(h)}$ with $j = 1, \ldots, 10$.

The resulting approximation accuracy of dense AMLS using the setting described above can be investigated in Table 2 which shows the relative errors from (32) and the corresponding error ratios. The approximation accuracy of dense AMLS in this configuration is very bad. The comparison — see Figure 3 where the approximated eigenfunctions associated to the 5 largest eigenvalues are displayed — between the eigenfunction approximations provided by $x_j^{(h)}$ of the discrete problem $(K^{(h)}, M^{(h)})$ and the eigenfunction approximations which are provided by dense AMLS via $\widehat{y}_j^{(h)}$ in (29) gives a possible explanation for this unsatisfying results. It seems that the eigenfunction approximations which are computed by dense AMLS are strongly varying from the exact ones especially at the region close to the interface of the subdomains $\Omega_1 = (0, \frac{1}{2})$ and $\Omega_2 = (\frac{1}{2}, 1)$. Further benchmarks showed that the approximation accuracy of dense AMLS could not be significantly improved just by increasing the number k_i of selected eigenpairs in task ($\mathbb{T}4$). Rather it seems that essential spectral information of the subproblems is missing in order to well approximate the sought eigenfunctions of the global problem $(K^{(h)}, M^{(h)})$.

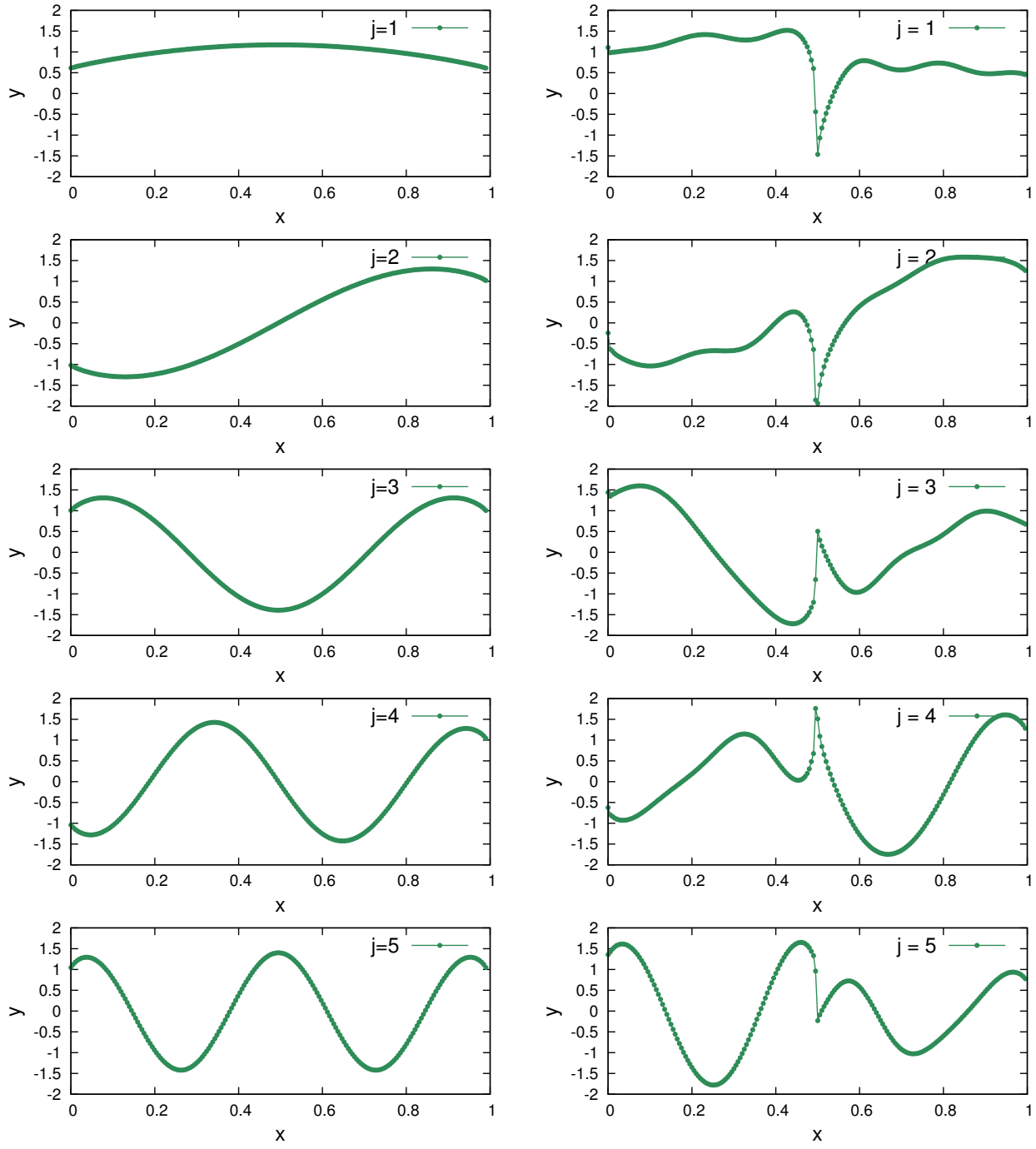


Figure 3: Approximations for the eigenfunctions u_j of problem (30) for $j = 1, \ldots, 5$: On the left-hand side are the approximations provided by the discrete eigenvectors $x_j^{(h)}$ of $(K^{(h)}, M^{(h)})$ where the underlying mesh width h leads to a discretisation with 200 DOF, and on the right-hand side are the approximations that are provided by the Ritz-vectors $\widehat{y}_j^{(h)}$ of the dense AMLS method. The function values of the eigenfunction approximations at the nodal points, which are provided by $x_j^{(h)}$ and $\widehat{y}_j^{(h)}$, have been connected by lines for a better visibility.

task		matrix operations of combined dense AMLS	
(ℂT1)	partition matrices K and M	apply geometric bisection reordering as done in (23)	apply geometric bisection reordering as done in (35)
(ℂT2)	block diagonalise K	$K = L\,\widetilde{K}\,L^T$	$K = L\,\widetilde{K}\,L^T$
(ℂT3)	transform M	$\widetilde{M} = L^{-1}\,M\,L^{-T}$	$\widetilde{M} = L^{-1}\,M\,L^{-T}$
(ℂT4)	compute partial eigensolutions	$K_{11}\,\widetilde{S}_1 = M_{11}\,\widetilde{S}_1\,\widetilde{D}_1$ $\widetilde{K}_{22}\,\widetilde{S}_2 = \widetilde{M}_{22}\,\widetilde{S}_2\,\widetilde{D}_2$	$K_{11}\,\widetilde{S}_1 = M_{11}\,\widetilde{S}_1\,\widetilde{D}_1$ $\widetilde{K}_{22}\,\widetilde{S}_2 = \widetilde{M}_{22}\,\widetilde{S}_2\,\widetilde{D}_2$
(ℂT5)	a) compute subspaces	$Z := \operatorname{diag}\left[\widetilde{S}_1, \widetilde{S}_2\right]$ $Q_A := L^{-T}Z \in \mathbb{R}^{N\times\bar{k}_A}$	$Z := \operatorname{diag}\left[\widetilde{S}_1, \widetilde{S}_2\right]$ $Q_B := L^{-T}Z \in \mathbb{R}^{N\times\bar{k}_B}$
	b) combine subspaces	$Q := [Q_A, Q_B] \in \mathbb{R}^{N\times\bar{k}}$ with $\bar{k} \le \bar{k}_A + \bar{k}_B$ (note: linear dependent columns in Q have to be removed)	
(ℂT6)	compute the reduced matrices	$\widehat{K} := Q^T\,K\,Q \in \mathbb{R}^{\bar{k}\times\bar{k}}$ and $\widehat{M} := Q^T\,M\,Q \in \mathbb{R}^{\bar{k}\times\bar{k}}$	
(ℂT7)	solve the reduced eigenvalue problem	$\widehat{K}\,\widehat{x}_j = \widehat{\lambda}_j\,\widehat{M}\,\widehat{x}_j$ for $j = 1, \ldots, n_{\text{es}}$	
(ℂT8)	transform eigenvectors	$\widehat{y}_j := Q\,\widehat{x}_j$ for $j = 1, \ldots, n_{\text{es}}$	

Table 3: Overview of the combined dense AMLS method to compute eigenpair approximations $(\widehat{\lambda}_j, \widehat{y}_j)$ for the problem (K, M).

Furthermore, it is noted that in general the eigenpair approximations, which are computed by dense AMLS, differ from each other if in dense AMLS instead of (23) the changed matrix partitioning

$$K = \begin{matrix} \\ \Omega_2 \\ \Omega_1 \end{matrix}\!\!\begin{matrix} \begin{matrix} \Omega_2 & \Omega_1 \end{matrix} \\ \begin{bmatrix} K_{22} & K_{12} \\ K_{21} & K_{11} \end{bmatrix} \end{matrix} \quad \text{and} \quad M = \begin{matrix} \\ \Omega_2 \\ \Omega_1 \end{matrix}\!\!\begin{matrix} \begin{matrix} \Omega_2 & \Omega_1 \end{matrix} \\ \begin{bmatrix} M_{22} & M_{12} \\ M_{21} & M_{11} \end{bmatrix} \end{matrix} \tag{35}$$

is applied. The reason for this, in simple terms, is that when matrix partitioning (23) is used then spectral information is transferred from Ω_1 to the subproblem associated to subdomain Ω_2, and when matrix partitioning (35) is used instead then spectral information is transferred from Ω_2 to the subproblem associated to Ω_1.

As already noted in Remark 4.2 i) the dense AMLS method can be interpreted as a Rayleigh-Ritz projection using the approximation subspace which is spanned by the columns of the matrix $Q := L^{-T}Z$. This interpretation of dense AMLS motivates to apply a Rayleigh-Ritz projection for the eigenvalue problem (K, M) where the associated approximation subspace is spanned by the columns of the matrix $Q_A := L^{-T}Z$ which is obtained by applying tasks (𝕋1)–(𝕋4) of dense AMLS using matrix partitioning (23) and by the columns of the matrix $Q_B := L^{-T}Z$ which is obtained by applying tasks (𝕋1)–(𝕋4) of dense AMLS using matrix partitioning (35). In the following this approach is referred to as *combined dense AMLS*. An overview of all necessary operations of combined

j	$\widehat{\delta}_j^{(h)}$	$\delta_j^{(h)}$	$\widehat{\delta}_j^{(h)}/\delta_j^{(h)}$
1	9.85e-6	3.67e-6	2.68e+0
2	2.89e-5	2.74e-5	1.06e+0
3	1.08e-4	9.70e-5	1.11e+0
4	2.12e-4	2.02e-4	1.05e+0
5	3.79e-4	3.52e-4	1.08e+0
6	5.44e-4	5.38e-4	1.01e+0
7	7.94e-4	7.68e-4	1.03e+0
8	1.05e-3	1.03e-3	1.01e+0
9	1.38e-3	1.34e-3	1.03e+0
10	1.69e-3	1.68e-3	1.00e+0
11	2.23e-3	2.07e-3	1.08e+0
12	5.05e-3	2.49e-3	2.03e+0
13	3.16e-2	2.95e-3	1.07e+1
14	9.32e-2	3.45e-3	2.70e+1
15	2.09e-1	3.99e-3	5.24e+1
16	4.91e-1	4.56e-3	1.08e+2
17	4.80e-1	5.17e-3	9.29e+1
18	5.37e-1	5.82e-3	9.22e+1
19	6.09e-1	6.50e-3	9.36e+1
20	8.84e-1	7.22e-3	1.22e+2

Table 4: Relative errors of combined dense AMLS $\widehat{\delta}_j^{(h)}$ versus relative discretisation errors $\delta_j^{(h)}$ for the eigenvalue approximation.

dense AMLS is given in Table 3 where the different tasks of the method are denoted by ($\mathbb{CT}1$)–($\mathbb{CT}8$). It has to be noted that in task ($\mathbb{CT}5$) b) the matrix $Q := [Q_A, Q_B]$ possibly has to be orthogonalised (e.g., by a QR-factorisation) in order to eliminate linear dependent columns in Q which are not wanted when the reduced eigenvalue problem is solved in task ($\mathbb{CT}7$).

Using the same parameter setting which has been used for the benchmarks of dense AMLS, the combined dense AMLS method leads to a reduced eigenvalue problem of size $\bar{k} = 10 + 10 = 20$. The resulting accuracy of combined dense AMLS for the eigenvalue approximation is displayed in Table 4. The benchmarks show that inequality (34) is fulfilled for $n_{\text{es}} = 12$, i.e., the approximation error of the combined dense AMLS method is of the same order as the discretisation error for $n_{\text{es}} = 12$. Furthermore, Figure 4 shows that the eigenfunction approximations provided by combined dense AMLS are nearly identical with the approximations provided by the discrete eigenvectors $x_j^{(h)}$ of $(K^{(h)}, M^{(h)})$.

The numerical results show that the combined dense AMLS method is potentially well suited for the solution of problem (6) respectively (1). However, up to this point the combined dense AMLS method is expensive. For example, the computation of tasks ($\mathbb{CT}2$) and task ($\mathbb{CT}3$) cause costs of the order $\mathcal{O}(N^3)$. To make out of combined dense AMLS an efficient eigensolver major improvements have to be made which are described in the following section.

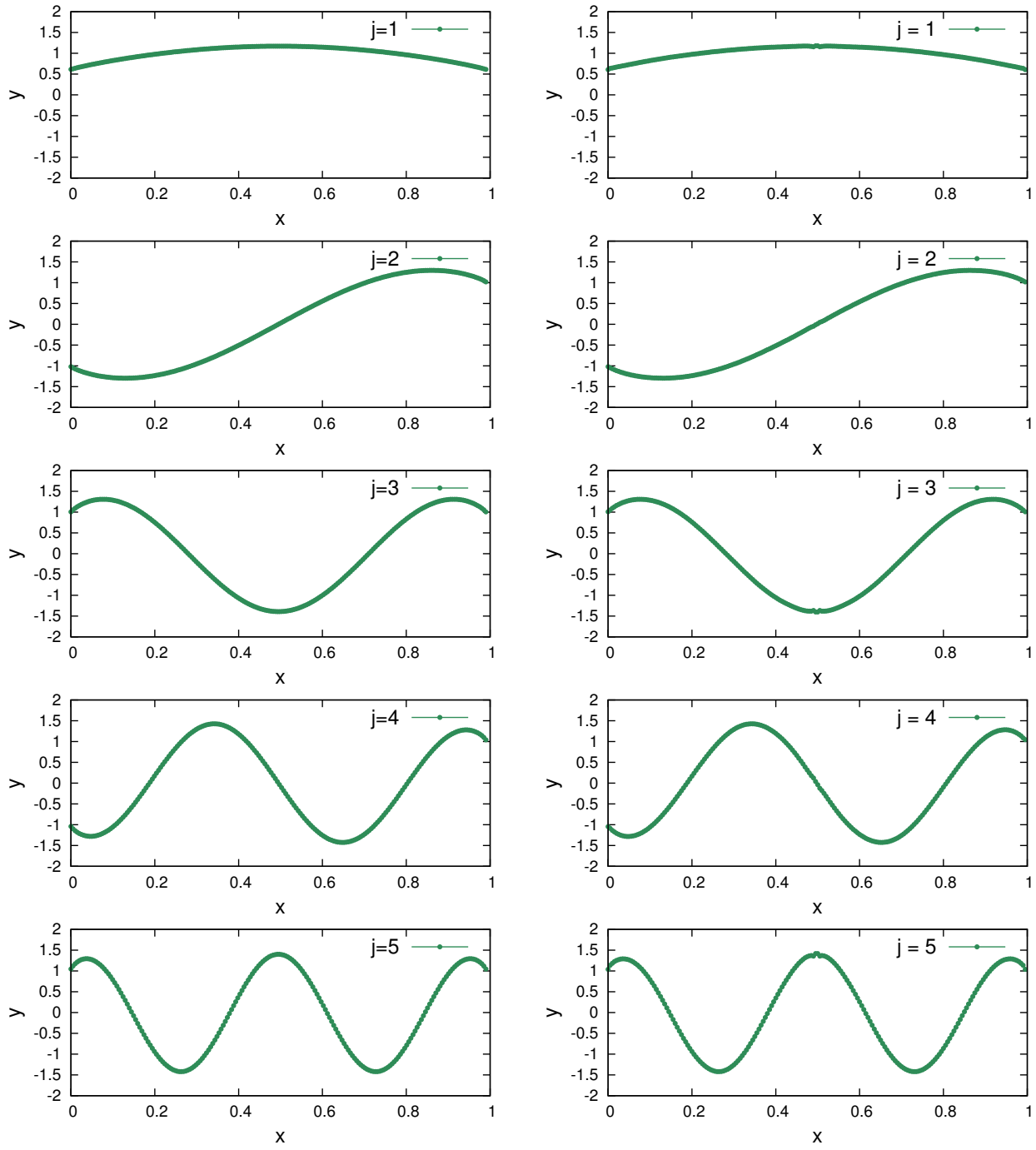


Figure 4: Approximations for the eigenfunctions u_j of problem (30) for $j = 1, \ldots, 5$: On the left-hand side are the approximations provided by the discrete eigenvectors $x_j^{(h)}$ of $(K^{(h)}, M^{(h)})$ where the underlying mesh width h leads to a discretisation with 200 DOF, and on the right-hand side are the approximations that are provided by the Ritz-vectors $\widehat{y}_j^{(h)}$ of the combined dense AMLS method.

6 Recursive Approach and Use of Hierarchical Matrices

There are two fundamental improvements of the combined dense AMLS method which will accelerate the computational time of the method dramatically. These improvements are motivated by the results presented in [13, 14], which show that the computational performance of classical AMLS in the context of elliptic PDE eigenvalue problems can be significantly improved by using a recursive approach and by using the concept of hierarchical matrices. The two fundamental improvements of the combined dense AMLS method are described in the following.

Recursive Approach

To improve the computational efficiency of the combined dense AMLS method, it is proposed to apply combined dense AMLS recursively in task ($\mathbb{CT}4$). More precisely, it is proposed to apply combined dense AMLS recursively for the solution of the subproblems (K_{11}, M_{11}) and $(\widetilde{K}_{22}, \widetilde{M}_{22})$ until the size N_i $(i = 1, 2)$ of the subproblems is small enough — e.g., smaller than a given threshold $N_{\min}^{\text{AMLS}} \in \mathbb{N}$ — to be solved easily by a direct solver. The depth of the recursion $l_{\text{rec}} \in \mathbb{N}$ depends on the size N of the problem (K, M) and is described by

$$N \approx 2^{l_{\text{rec}}} N_{\min}^{\text{AMLS}} \;\Rightarrow\; l_{\text{rec}} \sim \log N. \tag{36}$$

Usage of Hierarchical Matrices

The dense matrix operations in task ($\mathbb{CT}2$) and task ($\mathbb{CT}3$) of combined dense AMLS can be performed much more efficiently using the so-called hierarchical matrices (short $\mathcal{H}$-matrices). $\mathcal{H}$-matrices [21, 22] are matrices which basically consist of large low-rank submatrices and few small dense submatrices. This special matrix format is very well suited to approximate dense matrices which, e.g., arise in the context of the discretisation of partial differential and integral equations [4, 12, 16]. The basic idea of $\mathcal{H}$-matrices is to reorder the rows and columns of a matrix in a such way that certain submatrices can be approximated or represented by low rank matrices. Using the concept of low rank approximation a fully populated but data-sparse matrix of size $N \times N$ can be represented using only $\mathcal{O}(N \log^{\alpha} N)$ data instead of storing N^2 entries where $\alpha \in \{1, \ldots, 4\}$ (cf. [15, 17]). Most importantly, the concept of low rank approximation allows the $\mathcal{H}$-matrices to perform exact matrix-vector multiplication and approximated matrix(-matrix) operations (e.g., addition, multiplication, inversion, LU-factorisation) in almost linear complexity $O(N \log^{\alpha} N)$.

To use the fast $\mathcal{H}$-matrix arithmetic in task ($\mathbb{CT}2$) and task ($\mathbb{CT}3$) of combined dense AMLS, the dense stiffness matrix K and the sparse mass matrix M have first to be approximated by $\mathcal{H}$-matrices. How this is done is briefly described in the following for the matrix K. The same approach has to be used for the matrix M. To see more details it is referred, e.g., to [14, 15].

Since matrix $K \in \mathbb{R}^{N \times N}$ results from a finite element discretisation each row and column index $i \in I := \{1, \ldots, N\}$ is associated with a basis function $\varphi_i^{(h)}$ of the underlying finite element space V_h. The support of each index set $t \subset I$ is defined by

$$\Omega_t := \bigcup_{i \in t} \operatorname{supp}(\varphi_i^{(h)}),$$

and correspondingly each submatrix

$$K|_{s \times t} := (K_{ij})_{i \in s, j \in t} \qquad \text{with } s, t \subset I$$

of K is associated with geometry information. Based on the geometric separation of the supports Ω_s and Ω_t certain subblocks $s \times t \subset I \times I$ can be identified that allow a low rank approximation of the

respective submatrices $K|_{s\times t}$. More precisely, submatrices $K|_{s\times t}$ whose index sets s and t fulfil the so-called *admissibility condition*

$$\min\big\{\operatorname{diam}(\Omega_s), \operatorname{diam}(\Omega_t)\big\} \leq \eta \operatorname{dist}(\Omega_s, \Omega_t) \tag{37}$$

are well suited for a low rank approximation (cf. [15]). The quantities

$$\operatorname{diam}(\Omega_t) := \max\big\{\|x-y\|_2 \ : \ x, y \in \Omega_t\big\},$$

$$\operatorname{dist}(\Omega_s, \Omega_t) := \min\big\{\|x-y\|_2 \ : \ x \in \Omega_s, y \in \Omega_t\big\}$$

are the diameter and the distance of the supports Ω_s and Ω_t, and the parameter $\eta > 0$ controls the number of admissible subblocks $s \times t$ and is typically set to $\eta = 1$ (see, e.g., [15]).

Subblocks $s \times t$ of $I \times I$ which fulfil the admissibility condition (37) are called in the following *admissible* and the corresponding submatrices $K|_{s\times t}$ are approximated by the so-called R(k)-matrices.

Definition 6.1 (R(k)-matrix) *Let $k, m, n \in \mathbb{N}_0$. A matrix $R \in \mathbb{R}^{n\times m}$ is called* R(k)*-matrix if it is factorised by*

$$R = AB^T \quad \text{with suitable matrices} \quad A \in \mathbb{R}^{n\times k} \text{ and } B \in \mathbb{R}^{m\times k}. \tag{38}$$

The representation of an R(k)-matrix $R \in \mathbb{R}^{n\times m}$ in factorised form (38) is much cheaper than in full-matrix representation when the rank k is small compared to n and m because only $k(n+m)$ entries have to be stored instead of $n\,m$. Furthermore, when k is small the product and the sum of R(k)-matrices can be evaluated much more efficiently than in full-matrix representation.

To exploit the low rank approximation property of submatrices $K|_{s\times t}$ fulfilling (37) the row and column indices of K have to be reordered. For this purpose the index set I is divided according to a geometric bisection of its support Ω_I into two disjoint index sets $s, t \subset I$ with $I = s \dot\cup t$. In this context we denote s and t as the *sons* of I and $S(I) := \{s, t\}$ as the *set of sons* of I. This geometric bisection of I is applied recursively to the son index sets until the cardinality of an resulting index set falls below some given threshold $n_{min} \in \mathbb{N}$. This recursive geometric bisection of the index set I results in a disjoint partition of I where the obtained subsets of the partitioning tend to be geometrically separated (see, e.g., [15] for details). Finally, the row and column indices of the matrix K are reordered correspondingly to the performed partitioning of I.

Given the partitioning of I and the admissibility condition (37) the $\mathcal{H}$-matrix approximation of K is computed by applying Algorithm 1 to the block index set $I \times I$ and to the reordered matrix K. The resulting $\mathcal{H}$-matrix approximation is denoted by $K^{\mathcal{H}}$ in the following. Using Algorithm 1 the block index set $I \times I$ is recursively subdivided into subblocks $s \times t$ until the subblock gets admissible or the size of the subblock falls below the threshold n_{min}. Submatrices $K|_{s\times t}$ of admissible blocks $s \times t$ are approximated by R(k)-matrices and submatrices of inadmissible blocks are represented in the full-matrix format. To control in Algorithm 1 the approximation quality of the R(k)-matrix approximation the fixed rank k is replaced by an adaptive rank: Each submatrix $K|_{s\times t}$ corresponding to an admissible subblock $s \times t$ can be approximated by an R(k)-matrix R such that

$$\frac{\| K|_{s\times t} - R \|_2}{\| K|_{s\times t} \|_2} \leq \varepsilon \tag{39}$$

where $\varepsilon > 0$ is some arbitrary prescribed approximation accuracy and where the rank $k \in \mathbb{N}_0$ is as small as possible (cf. [15]).

Using the same approach, we compute the $\mathcal{H}$-matrix approximation $M^{\mathcal{H}}$ of M, i.e., first we reorder the row and column indices of M according to the partitioning of I, and then we apply Algorithm 1 to the block index set $I \times I$ and the reordered matrix M. The approximation errors $\|K - K^{\mathcal{H}}\|_2$ and $\|M - M^{\mathcal{H}}\|_2$ are controlled by the accuracy ε of the R(k)-matrix approximation in (39).

Algorithm 1 Computation of the $\mathcal{H}$-Matrix Approximation

procedure GETHMATRIXAPPROXIMATION(K, ε, n_{min}, $s \times t$)
  if $s \times t$ is admissible **then**
    $K^{\mathcal{H}}|_{s\times t}$:= R(k)-matrix approximation of $K|_{s\times t}$ with accuracy ε;
  else if $\min\{\#s, \#t\} \leq n_{min}$ **then**
    $K^{\mathcal{H}}|_{s\times t}$:= full-matrix representation of $K|_{s\times t}$; ▷ n_{min} affects the minimal size of 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**
      $K^{\mathcal{H}}|_{s'\times t'}$:=GETHMATRIXAPPROXIMATION(K, ε, n_{min}, $s' \times t'$);
    end for
  end if
end procedure

The $\mathcal{H}$-matrix approximations $K^{\mathcal{H}}$ and $M^{\mathcal{H}}$ of K and M can now be used to compute in task ($\mathbb{CT}2$) and ($\mathbb{CT}3$) of combined dense AMLS the transformed eigenvalue problem $(\widetilde{K}, \widetilde{M})$. Note that the $\mathcal{H}$-matrices $K^{\mathcal{H}}$ and $M^{\mathcal{H}}$ possess a recursive 2×2 block partitioning of the form

$$K^{\mathcal{H}} = \begin{bmatrix} K_{11}^{\mathcal{H}} & K_{12}^{\mathcal{H}} \\ K_{21}^{\mathcal{H}} & K_{22}^{\mathcal{H}} \end{bmatrix} \qquad \text{and} \qquad M^{\mathcal{H}} = \begin{bmatrix} M_{11}^{\mathcal{H}} & M_{12}^{\mathcal{H}} \\ M_{21}^{\mathcal{H}} & M_{22}^{\mathcal{H}} \end{bmatrix} \tag{40}$$

where $K_{ij}^{\mathcal{H}}, M_{ij}^{\mathcal{H}} \in \mathbb{R}^{N_i \times N_j}$ are either full-matrices, R(k)-matrices or $\mathcal{H}$-matrices which again have a 2×2 block partitioning. Because of the matrix partitioning (40) of $K^{\mathcal{H}}$ and $M^{\mathcal{H}}$ (which is based on a geometric bisection of the support Ω_I), there is no need to perform the domain substructuring and the associated matrix partitioning of task ($\mathbb{CT}1$) in combined dense AMLS. Using the fast $\mathcal{H}$-matrix arithmetic and the $\mathcal{H}$-matrix approximations $K^{\mathcal{H}}$ and $M^{\mathcal{H}}$, the block diagonalisation of K and the transformation of M can be computed very efficiently via

$$K^{\mathcal{H}} \approx L^{\mathcal{H}} \widetilde{K}^{\mathcal{H}} (L^{\mathcal{H}})^T \quad \text{and} \quad \widetilde{M}^{\mathcal{H}} \approx (L^{\mathcal{H}})^{-1} M^{\mathcal{H}} (L^{\mathcal{H}})^{-T} \tag{41}$$

in $\mathcal{O}(N \log^{\alpha} N)$. The corresponding algorithm for the computation of (41) is based on a recursive approach which is applied block-wise to the matrix structure (40), which exploits the R(k)-matrix representation of submatrices fulfilling (37), and which applies the inexpensive addition and multiplication of R(k)-matrices (see, e.g., [15] for details). The $\mathcal{H}$-matrix operations in (41) are performed not exactly but only approximatively, however, the approximation errors $\|\widetilde{K} - \widetilde{K}^{\mathcal{H}}\|_2$ and $\|\widetilde{M} - \widetilde{M}^{\mathcal{H}}\|_2$ can be controlled by the chosen accuracy ε in (39).

The improved version of combined dense AMLS with the recursive approach and the use of the fast $\mathcal{H}$-matrix arithmetic is simply called *dense $\mathcal{H}$-AMLS* in the following. It is proposed to use the $\mathcal{H}$-matrix software library HLIBpro [8, 24, 25] for the implementation of the method; and since dense $\mathcal{H}$-AMLS has much in common with the method described in [13], it is recommended to see [13, Section 8] for corresponding implementation issues.

Beside the number of sought eigenpairs n_{es} and the number of degrees of freedom N the computational costs of dense $\mathcal{H}$-AMLS depend on the chosen accuracy ε of the approximative $\mathcal{H}$-matrix operations in (41), and on the applied modal truncation in task ($\mathbb{CT}4$), i.e, the number of selected eigenvectors k_i in (16). Coarsening the accuracy ε of the $\mathcal{H}$-matrix operations and decreasing the number k_i of selected eigenvectors result in faster computations and reduced memory requirements of dense $\mathcal{H}$-AMLS. On the other side the accuracy ε has to be fine enough and the size k_i ($i = 1, 2$) has

to be large enough so that the error caused by the approximative $\mathcal{H}$-matrix arithmetic and the error caused by the modal truncation are of the same order as the discretisation error. If this is the case the approximation quality of dense $\mathcal{H}$-AMLS competes with the approximation quality of a classical approach [cf. Remark 3.1 i)]. Hence, the aim arises to select the parameters ε and k_i in such a way that dense $\mathcal{H}$-AMLS reaches the accuracy of a classical approach while the computational costs of $\mathcal{H}$-AMLS are reduced as much as possible.

In the following the computational costs of dense $\mathcal{H}$-AMLS are discussed in detail. The discussion is restricted to problems where $n_{\text{es}} \sim N^{\beta}$ eigenpairs are sought with some $\beta \in (0, 1/3]$. In task ($\mathbb{CT}4$) the mode selection strategy proposed by Remark 4.1 is applied, i.e., the eigenvectors of $(K_{11}^{\mathcal{H}}, M_{11}^{\mathcal{H}})$ and $(\widetilde{K}_{22}^{\mathcal{H}}, \widetilde{M}_{22}^{\mathcal{H}})$ associated to the k_i (i=1,2) smallest eigenvalues are computed where $k_i \in \mathcal{O}(N_i^{\beta})$, and thus it is guaranteed that the size $\bar{k}$ of the reduced problem in task ($\mathbb{CT}7$) is bounded by $\mathcal{O}(n_{\text{es}})$. Using this setting the costs of dense $\mathcal{H}$-AMLS can be summarised as follows:

- The computational costs for the matrix partitioning in ($\mathbb{CT}1$), respectively the costs for the computation of the $\mathcal{H}$-matrix approximations $K^{\mathcal{H}}$ and $M^{\mathcal{H}}$ are of the order $\mathcal{O}(N \log N)$. The computational costs for task ($\mathbb{CT}2$) and ($\mathbb{CT}3$) using the fast $\mathcal{H}$-matrix arithmetic are of the order $\mathcal{O}(N \log^{\alpha} N)$.

- In task ($\mathbb{CT}4$) four subproblems arise (cf. Table 1) of size $N_i \approx N/2$ (with $i = 1, 2$). In the case that the subproblems are small enough (i.e., it holds $N_i \leq N_{\min}^{\text{AMLS}}$) the subproblems are solved by a direct solver leading to costs of the order $\mathcal{O}(1)$, otherwise the subproblems are solved recursively by the dense $\mathcal{H}$-AMLS method.

- The computation of the matrices $Q_A = (L^{\mathcal{H}})^{-T} Z$ and $Q_B = (L^{\mathcal{H}})^{-T} Z$ in task ($\mathbb{CT}5$) can be performed efficiently by exploiting the fast $\mathcal{H}$-matrix times vector multiplication [backward substitution in $(L^{\mathcal{H}})^T$]. Since the number of columns of $Q_A \in R^{N \times \bar{k}_A}$ and $Q_B \in R^{N \times \bar{k}_B}$ is bounded by $\bar{k}_A, \bar{k}_B \in \mathcal{O}(N^{\beta})$, the computation of Q_A and Q_B can be performed in $\mathcal{O}(N^{\beta}\, N \log^{\alpha} N)$. The computation of the matrix $Q \in \mathbb{R}^{N \times \bar{k}}$ where $\bar{k} \leq \bar{k}_A + \bar{k}_B$, more precisely, the elimination of linear dependent columns in the matrix $[Q_A, Q_B]$ by orthogonalisation, leads to computational costs of the order $\mathcal{O}(N^{2\beta} N)$. Since $n_{\text{es}} \sim N^{\beta}$ the total costs of task ($\mathbb{CT}5$) can be bounded by

$$\mathcal{O}\big(\, N^{\beta}\, N \log^{\alpha} N \,\big) + \mathcal{O}\big(\, N^{2\beta}\, N \,\big) = \mathcal{O}\big(\, n_{\text{es}}\, N \log^{\alpha} N \,\big) + \mathcal{O}\big(\, n_{\text{es}}^2\, N \,\big).$$

- The $\mathcal{H}$-matrix structure of $K^{\mathcal{H}}$ and $M^{\mathcal{H}}$ can be exploited as well in task ($\mathbb{CT}6$) using the fast $\mathcal{H}$-matrix-vector multiplication for the computation of the reduced matrices $\widehat{K}, \widehat{M} \in \mathbb{R}^{\bar{k} \times \bar{k}}$: The multiplications $Q^T(K^{\mathcal{H}}Q)$ and $Q^T(M^{\mathcal{H}}Q)$ involve in total $2\bar{k}$ $\mathcal{H}$-matrix times vector multiplications with costs of the order $\mathcal{O}(N \log^{\alpha} N)$ plus $2\bar{k}^2$ scalar products of length N. Hence, the costs of ($\mathbb{CT}6$) sum up to

$$\mathcal{O}\big(\, \bar{k}\, N \log^{\alpha} N \,+\, \bar{k}^2\, N \,\big) \;\leq\; \mathcal{O}\big(\, n_{\text{es}}\, N \log^{\alpha} N \,+\, n_{\text{es}}^2\, N \,\big).$$

- Since the size $\bar{k}$ of the reduced problem $(\widehat{K}, \widehat{M})$ is bounded by $\bar{k} \in \mathcal{O}(n_{\text{es}})$ and since we aim at $n_{\text{es}} \in \mathcal{O}(N^{1/3})$ eigenpairs, we can use in task ($\mathbb{CT}7$) a dense linear algebra solver with cubic complexity for the solution of the reduced problem whereby the computational costs still remain in $\mathcal{O}(N)$.

- Finally, in task ($\mathbb{CT}8$) in total $N n_{\text{es}}$ scalar products of length $\bar{k}$ have to performed leading to costs of the order

$$\mathcal{O}\big(\, n_{\text{es}} \bar{k} N \,\big) \;\leq\; \mathcal{O}\big(\, n_{\text{es}}^2 N \,\big).$$

Hence, the computational costs of dense $\mathcal{H}$-AMLS — without the costs of the recursive calls of dense $\mathcal{H}$-AMLS in task ($\mathbb{CT}4$) — are bounded by $\mathcal{O}(n_{\text{es}} N \log^{\alpha} N + n_{\text{es}}^2 N)$. Note that when dense $\mathcal{H}$-AMLS is applied recursively to a problem of size N, that then on the next level of the recursion four new subproblems (cf. Table 1) arise of approximate size $N/2$ which are handled recursively by dense $\mathcal{H}$-AMLS. It follows that the overall computational costs of dense $\mathcal{H}$-AMLS, including the costs for the recursive calls, are bounded by

$$
\begin{aligned}
&\mathcal{O}\big(\, n_{\text{es}}\, N \log^{\alpha} N + n_{\text{es}}^2\, N \,\big) + \sum_{l=1}^{l_{\text{rec}}} 4^l\, \mathcal{O}\bigg(\Big(\frac{N}{2^l}\Big)^{\beta} \frac{N}{2^l} \log^{\alpha}\Big(\frac{N}{2^l}\Big) \;+\; \Big(\frac{N}{2^l}\Big)^{2\beta} \frac{N}{2^l} \bigg) \\
&= \mathcal{O}\big(\, n_{\text{es}}\, N \log^{\alpha} N + n_{\text{es}}^2\, N \,\big) + \sum_{l=1}^{l_{\text{rec}}} 2^l\, \mathcal{O}\bigg(\Big(\frac{N}{2^l}\Big)^{\beta} N \log^{\alpha} N \;+\; \Big(\frac{N}{2^l}\Big)^{2\beta} N \bigg) \\
&= \mathcal{O}\big(\, n_{\text{es}}\, N \log^{\alpha} N + n_{\text{es}}^2\, N \,\big) + \sum_{l=1}^{l_{\text{rec}}} 2^l\, \mathcal{O}\bigg(\frac{n_{\text{es}}}{2^{l\beta}} N \log^{\alpha} N \;+\; \frac{n_{\text{es}}^2}{2^{2l\beta}} N \bigg) \\
&= \mathcal{O}\big(\, n_{\text{es}}\, N \log^{\alpha} N \,\big) \sum_{l=0}^{l_{\text{rec}}} 2^{l(1-\beta)} \;+\; \mathcal{O}\big(\, n_{\text{es}}^2\, N \,\big) \sum_{l=0}^{l_{\text{rec}}} 2^{l(1-2\beta)} \\
&= \mathcal{O}\big(\, n_{\text{es}}\, N \log^{\alpha} N \,\big)\, \frac{2^{(l_{\text{rec}}+1)(1-\beta)} - 1}{2^{1-\beta} - 1} \;+\; \mathcal{O}\big(\, n_{\text{es}}^2\, N \,\big)\, \frac{2^{(l_{\text{rec}}+1)(1-2\beta)} - 1}{2^{1-2\beta} - 1} \\
&\overset{(36)}{=} \mathcal{O}\big(\, n_{\text{es}}\, N \log^{\alpha} N\, N^{(1-\beta)} \,\big) \;+\; \mathcal{O}\big(\, n_{\text{es}}^2\, N\, N^{(1-2\beta)} \,\big) \;=\; \mathcal{O}\big(\, N^2 \log^{\alpha} N + N^2 \,\big)
\end{aligned}
$$

We can sum up that the theoretical complexity of dense $\mathcal{H}$-AMLS is bounded[2] by $\mathcal{O}(N^2 \log^{\alpha} N + N^2)$. The costs of dense $\mathcal{H}$-AMLS are theoretically dominated by the costs of of the scalar products occurring in tasks ($\mathbb{CT}5$), ($\mathbb{CT}6$) and ($\mathbb{CT}8$) which are accumulating to $\mathcal{O}(N^2)$, and the costs of the $\mathcal{H}$-matrix times vector multiplications in tasks ($\mathbb{CT}5$) and ($\mathbb{CT}6$) which are accumulating to $\mathcal{O}(N^2 \log^{\alpha} N)$. However, since scalar products can be computed with peak performance on today's workstations and compute servers, it is expected that these operations will not dominate the overall costs of dense $\mathcal{H}$-AMLS in practice. It is expected as well that in practice the $\mathcal{H}$-matrix times vector multiplications are as well harmless since the logarithms and constants involved in the $\mathcal{H}$-matrix vector multiplications are much smaller than for the $\mathcal{H}$-matrix operations performed in task ($\mathbb{CT}2$) and task ($\mathbb{CT}3$). Overall, it is expected that in practice the costs of dense $\mathcal{H}$-AMLS are dominated by the costs of the $\mathcal{H}$-matrix operations in task ($\mathbb{CT}2$) and ($\mathbb{CT}3$) which, however, accumulate as well, due to the recursive calls, to costs of the order $\mathcal{O}(N^2 \log^{\alpha} N)$. But possibly these costs can be bounded by some approach which limits the number of needed problem transformations [task ($\mathbb{CT}2$) and ($\mathbb{CT}3$)] in the recursive calls. In [13, Section 8] an approach is described for a quite similar method which allows to replace all needed problem transformations of the recursive calls by only one global problem transformation. Furthermore, the very high efficiency of the method described in [13, 14] (which combines in the context of elliptic PDE eigenvalue problems the classical AMLS method with a recursive approach and the usage of the $\mathcal{H}$-matrices) towards classical approaches motivates that the very similar dense $\mathcal{H}$-AMLS method reaches in practice possibly the same superior efficiency.

7 Conclusion

While domain decomposition methods, like the very efficient AMLS method, are available for the solution of elliptic PDE eigenvalue problems, domain decomposition techniques are not known for

[2] Note that the upper bound $\mathcal{O}(N^2 \log^{\alpha} N + N^2)$ for the computational costs of dense $\mathcal{H}$-AMLS computing the $n_{\text{es}} \sim N^{\beta}$ smallest eigenpairs is obtained for all $\beta \in (0, 1)$, i.e., also in the case when it holds $\beta > 1/3$.

the solution integral equation eigenvalue problems. To the best of the author's knowledge this paper introduces the very first domain decomposition method for integral equation eigenvalue problems. The new method, which is motivated by the classical AMLS method, decomposes a global problem with N degrees of freedom into four subproblems each with approximately $N/2$ degrees of freedom. The eigensolutions of the four subproblems are then used to form a subspace which approximates the sought eigensolutions of the global problem. This domain decomposition technique is called combined dense AMLS method and shows very promising results concerning the approximation quality. To improve the computational efficiency of the combined dense AMLS method, it is proposed to use a recursive approach and the fast $\mathcal{H}$-matrix arithmetic for the needed problem transformation. The theoretical computational complexity of this approach is of the order $\mathcal{O}(N^2 \log^\alpha N)$, however, motivated by the results for a quite similar method (cf. [13, 14]) it is expected that the method reaches in practice a much better computational efficiency. Furthermore, the promising results of the combined dense AMLS method show that domain decomposition techniques are also applicable to integral equation eigenvalue problems and motivate to refine the domain decomposition techniques of combined dense AMLS. Possibly the combined dense AMLS method can be refined in such a way that the global problem is decomposed into only two subproblems instead of four which might decrease the computational complexity.

References

[1] E. Anderson, Z. Bai, C. Bischof, L. S. Blackford, J. Demmel, J. J. Dongarra, J. Du Croz, S. Hammarling, A. Greenbaum, A. McKenney, and D. Sorensen. *LAPACK Users' Guide (Third Ed.)*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1999.

[2] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst, editors. *Templates for the solution of algebraic eigenvalue problems*, volume 11 of *Software, Environments, and Tools*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000. A practical guide.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

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

[9] F. Bourquin. Analysis and comparison of several component mode synthesis methods on one-dimensional domains. *Numerische Mathematik*, 58:11–34, 1990.

[10] F. Bourquin. Component mode synthesis and eigenvalues of second order operators: Discretization and algorithm. *Mathematical Modeling and Numerical Analysis*, 26:385–423, 1992.

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

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

[13] P. Gerds. *Solving an elliptic PDE eigenvalue problem via automated multi-level substructuring and hierarchical matrices*. Ph.d. thesis, RWTH Aachen University, 2017.

[14] P. Gerds and L. Grasedyck. Solving an elliptic pde eigenvalue problem via automated multi-level substructuring and hierarchical matrices. *Computing and Visualization in Science*, 16(6):283–302, 2015.

[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. *Integralgleichungen : Theorie und Numerik*, volume 68 of *Leitfden der angewandten Mathematik und Mechanik*. B.G. Teubner, Stuttgart, 1989.

[20] W. Hackbusch. *Integral equations : theory and numerical treatment*, volume 120 of *International series of numerical mathematics*. Birkhuser, Basel, 1995.

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

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

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

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

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

[26] 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.

[27] C. L. Lawson, R. J. Hanson, D. R. Kincaid, and F. T. Krogh. Basic linear algebra subprograms for fortran usage. *ACM Trans. Math. Softw.*, 5(3):308–323, Sept. 1979.

[28] 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.

[29] 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.

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

[31] D. S. Watkins. Qr-like algorithms for eigenvalue problems. *Journal of Computational and Applied Mathematics*, 123(1):6783, 2000. Numerical Analysis 2000. Vol. III: Linear Algebra.

[32] 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.