



DFG-Research Center MATHEON
Mathematics for Key Technologies

**Adaptive Numerical Solution of Eigenvalue
Problems arising from Finite Element Models.
AMLS vs. AFEM**

Christoph Conrads Volker Mehrmann Agnieszka Międlar

Preprint 2015/02

MATHEON preprint

<http://opus4.kobv.de/opus4-mattheon>

Preprint 2015/02

February 2015

Adaptive Numerical Solution of Eigenvalue Problems arising from Finite Element Models. AMLS vs. AFEM

C. Conrads, V. Mehrmann, and A. Międlar

ABSTRACT. We discuss adaptive numerical methods for the solution of eigenvalue problems arising either from the finite element discretization of a partial differential equation (PDE) or from discrete finite element modeling. When a model is described by a partial differential equation, the adaptive finite element method starts from a coarse finite element mesh which, based on a posteriori error estimators, is adaptively refined to obtain eigenvalue/eigenfunction approximations of prescribed accuracy. This method is well established for classes of elliptic PDEs, but is still in its infancy for more complicated PDE models. For complex technical systems, the typical approach is to directly derive finite element models that are discrete in space and are combined with macroscopic models to describe certain phenomena like damping or friction. In this case one typically starts with a fine uniform mesh and computes eigenvalues and eigenfunctions using projection methods from numerical linear algebra that are often combined with the algebraic multilevel substructuring to achieve an adequate performance. These methods work well in practice but their convergence and error analysis is rather difficult. We analyze the relationship between these two extreme approaches. Both approaches have their pros and cons which are discussed in detail. Our observations are demonstrated with several numerical examples.

2010 *Mathematics Subject Classification.* 65F15, 65N25, 65N30.

Technische Universität Berlin, Institut für Mathematik, MA 4-5, Straße des 17. Juni 136, 10623 Berlin, Germany. conrads@math.tu-berlin.de.

Technische Universität Berlin, Institut für Mathematik, MA 4-5, Straße des 17. Juni 136, 10623 Berlin, Germany. mehrmann@math.tu-berlin.de. Research supported in the framework of MATHEON project *D-OT3, Adaptive finite element methods for nonlinear parameter-dependent eigenvalue problems in photonic crystals* supported by Einstein Foundation Berlin.

ANCHP – MATHICSE, École Polytechnique Fédérale de Lausanne, Bâtiment MA, Station 8, CH-1015 Lausanne, Switzerland. agnieszka.miedlar@epfl.ch and Technische Universität Berlin, Institut für Mathematik, MA 4-5, Straße des 17. Juni 136, 10623 Berlin, Germany. mehrmann@math.tu-berlin.de. Research supported in the framework of MATHEON project *D-OT3, Adaptive finite element methods for nonlinear parameter-dependent eigenvalue problems in photonic crystals* supported by Einstein Foundation Berlin, within a DFG Research Fellowship under the DFG GEPRIS Project *Adaptive methods for nonlinear eigenvalue problems with parameters* and by the Chair of Numerical Algorithms and High-Performance Computing (ANCHP), Mathematics Institute of Computational Science and Engineering (MATHICSE), École Polytechnique Fédérale de Lausanne.

1. Introduction

Eigenvalue problems associated with complex mathematical models described by partial differential equations (PDEs) or very large finite element models arise in a number of applications ranging from quantum mechanical models, design of periodic structures for wave-guides, structural mechanics, stability analysis in dynamical systems, as well as model reduction. A typical real world example that was recently considered in [33] is the analysis and treatment of disc brake squeal. This phenomenon arises from self-excited vibrations caused by a flutter-type instability originating from friction forces at the pad-rotor interface [2] of the brake. Since a full atomistic modeling via the Langevin equation [64] is computationally infeasible, a commonly used approach in practice is to employ macro-scale approximations via multibody dynamics and finite element modeling (FEM) with very fine uniform meshes, see e.g. [45, 56]. Using macroscopic models of material damping, and friction forces, one obtains as model equations for the finite element coefficients of the position variables $Q = \sum_{i=1}^n Q_i(t)\varphi_i(x)$, the dynamical system

$$(1) \quad M\ddot{Q} + C\dot{Q} + KQ = f,$$

where $M, C, K \in \mathbb{R}^{n,n}$ are large scale mass, damping and stiffness matrices, respectively, and f is an external force. For finite element (FE) models of rotating machinery, the matrices C and K are typically nonsymmetric to incorporate gyroscopic and circulatory forces, and they depend on various parameters that include model operating conditions, material conditions as well as the rotational speed of the disc. For self-excited vibrations one also includes the excitation force via a nonsymmetric term added to the stiffness matrix. Furthermore, the mass matrix is often singular, often due to mass lumping, or due to the explicit algebraic equations which constrain the dynamics of the system.

System (1) is a (space) discrete FE model and typically, since the macroscopic approximations do not hold in the continuous limit for mesh size equal to zero, a corresponding continuous model in the form of a PDE is not available. If the system would be valid in the limit, then it would be a hyperbolic PDE

$$(2) \quad \ddot{U} + c\dot{U} + k \Delta U = f,$$

in a domain $\Omega \in \mathbb{R}^d, d = 1, 2, \dots$, with given parameter dependent functions c, k , together with boundary conditions. We discuss here Dirichlet conditions $U = 0$ on the boundary $\partial\Omega$, the case of general boundary conditions can be reduced to this case [11].

Choosing the ansatz $Q = \exp(\mu t)\mathbf{q}$ or $U = \exp(\mu t)u$, respectively, then yields eigenvalue problems. In the (space) discrete case this is the finite dimensional quadratic eigenvalue problem:

Determine $\mu \in \mathbb{C}$ and a nonzero $\mathbf{q} \in \mathbb{C}^n$ such that

$$(3) \quad L(\mu)\mathbf{q} := (\mu^2 M + \mu C + K)\mathbf{q} = 0.$$

A finite dimensional eigenvector $\mathbf{q} = [q_i]_{i=1}^n \in \mathbb{C}^n$ associated with an eigenvalue μ is then the coordinate vector of coefficients for the eigenfunction $q = \sum_{i=1}^n q_i \psi^i(x)$ in the FE basis $\{\psi^i(x)\}_{i=1}^n$, which has been used to generate the FE model.

In the (space) continuous case one obtains an infinite dimensional quadratic eigenvalue problem.

Determine $\mu \in \mathbb{C}$ and a nonzero function u in an appropriate function space V such that

$$(4) \quad \mathcal{L}(\mu)u := (\mu^2 + c\mu + k\Delta)u = 0 \text{ in } \Omega, \quad u = 0 \text{ on } \partial\Omega.$$

In order to solve the continuous problem (4) one discretizes the variational form of the problem using a Galerkin approach. If an appropriate n_h dimensional FE subspace $V_h \subset V$ spanned by a FE basis $\{\varphi_h^{(i)}(x)\}_{i=1}^{n_h}$ is chosen, then (4) takes the form of the finite dimensional quadratic eigenvalue problem

$$(5) \quad \mathcal{L}_h(\mu)u_h := (\mu_h^2 M_h + C_h \mu_h + K_h)u_h = 0,$$

and the eigenfunction u_h is represented as $u_h = \sum_{i=1}^{n_h} u_{h,i} \varphi_h^{(i)}(x)$, with a coordinate vector $\mathbf{u}_h = [u_{h,i}]_{i=1}^{n_h} \in \mathbb{C}^{n_h}$.

The two discrete finite dimensional problems (3) and (5) are very similar in nature. If a PDE model is available, then using the same basis functions $\psi^{(i)} = \varphi_h^{(i)}$, $i = 1, 2, \dots, n_h$, on the same uniform grid without exploiting any macroscopic model approximations, would result in strongly related or even the same matrix coefficients.

In the continuous case, for some PDE operators, a priori and a posteriori error estimates can be obtained that allow to assess the quality of the solution and allow for adaptive grid refinement. In the discrete case, however, such estimates are typically not available, and error control has to be based on algebraic techniques and comparisons with experiments. However, in both the discrete and continuous case, once approximations $\tilde{\mu}$ of an eigenvalue μ and $\tilde{\mathbf{q}}(x) = \sum_{i=1}^n \tilde{q}_i \psi^{(i)}(x)$, respectively, $\tilde{u}_h(x) = \sum_{i=1}^{n_h} \tilde{u}_{h,i} \varphi_h^{(i)}(x)$ to the associated eigenfunctions have been determined, then eigenvalue residuals can be formed

$$(6) \quad r := (\tilde{\mu}^2 M + \tilde{\mu} C + K)\tilde{\mathbf{q}},$$

$$(7) \quad r_h(x) := (\tilde{\mu}_h^2 M_h + C_h \tilde{\mu}_h + K_h)\tilde{u}_h,$$

respectively. Using backward error analysis [53], it follows that if the stability constant of the associated eigenvalue/eigenfunction pair (μ_h, u_h) , respectively the condition number of the eigenvalue/eigenvector pair (μ, \mathbf{q}) are not too large, i.e., small perturbations in the model do not lead to large perturbations in computed eigenvalue/eigenfunction approximations, then these residuals can be used to estimate the associated errors. This analysis will be discussed in Section 4.

There are essentially two extreme approaches to compute a specific eigenvalue/eigenvector or eigenvalue/eigenfunction pair. In the continuous case, when a priori and a posteriori error estimates are available, then one can apply the adaptive finite element method (AFEM), and, at least in some special cases, prove its reliability and efficiency [15]. Starting from a sufficiently fine initial mesh, AFEM uses local error estimates to adaptively refine the mesh, so that the resulting error in the finite element approximation is within a given tolerance. However, the analytic background of AFEM requires the presence of an associated PDE model which, as discussed before, is not always available. Furthermore, the method is currently restricted to some very special problem classes of elliptic problems with real or purely imaginary eigenvalues. Numerical methods that use AFEM for some more complex problems were discussed in [16, 38, 61], but no detailed analysis is available for problems with complex eigenvalues or eigenvalues with Jordan blocks. For multiple real eigenvalues of symmetric problems recently there has been analysis in [28,

29, 72]. In contrast to this, in (3) the only available data are matrices associated with the fine mesh. A very common method that is frequently used in structural engineering is the automated multilevel substructuring method (AMLS) [9, 42]. Starting from (3) or (5) associated with the fine mesh, AMLS uses an algebraic substructuring technique (component mode synthesis [60]), local computations of eigenvalues and eigenvectors for algebraically constructed substructures in (3), as well as projections, to obtain a small projected eigenvalue problem which then can be solved by standard, dense eigenvalue methods [7, 36]. This algebraic approach is very flexible and can in principle be extended to treat damping terms as well as complex and multiple eigenvalues. However, the theoretical analysis of the method is rather limited [24, 37, 71], and the results may not always be satisfactory, see Section 6.

In this work we discuss a common basis for both the AFEM and the AMLS method. We compare these concepts and point out their advantages and disadvantages. For a direct analytical comparison, we restrict ourselves to consider (3) and (5) with symmetric positive definite mass and definite stiffness matrices and discard damping or other parts of the model, see Section 2. In this situation, setting $\lambda := -\mu^2$, all the eigenvalues are real and there exist orthonormal sets of associated eigenvectors/eigenfunctions. In Section 3 we introduce the automated multilevel substructuring method (AMLS) and discuss its properties and several implementation details. Section 4 is dedicated to the adaptive finite element method (AFEM) and a recently developed variant called AFEMLA. We compare both methods, provide some common ground and discuss some of their advantages and disadvantages in Section 5. Section 6 illustrates our observations with several numerical examples.

2. A model problem for comparison

In order to compare and relate the AFEM and the AMLS method, we use a simple elliptic PDE eigenvalue problem.

Determine $\lambda \in \mathbb{R}$ and $u \in V := H_0^1(\Omega)$ (the space of functions that vanish on the boundary and have a first derivative that is Lebesgue integrable) such that

$$(8) \quad \begin{aligned} \mathcal{L}u &= \lambda u && \text{in } \Omega \\ u &= 0 && \text{on } \partial\Omega, \end{aligned}$$

where $\Omega \in \mathbb{R}^d$, $d = 1, 2, \dots$ is a bounded, polyhedral Lipschitz domain and $\partial\Omega$ is its boundary.

Here

$$\mathcal{L}u(x) = -\operatorname{div}(A(x)\nabla u(x)),$$

and A is a real symmetric positive definite matrix, so that \mathcal{L} is self-adjoint and elliptic. Introducing the bilinear forms

$$\begin{aligned} a : V \times V &\rightarrow \mathbb{R}, & a(u, v) &:= \int_{\Omega} (\nabla u)^T A(x) \nabla v \, dx, \\ b : V \times V &\rightarrow \mathbb{R}, & b(u, v) &:= \int_{\Omega} uv \, dx, \end{aligned}$$

then one has the variational form of problem (8):

Determine $\lambda \in \mathbb{R}$ and $u \in V$ such that

$$(9) \quad a(u, v) = \lambda b(u, v) \quad \text{for all } v \in V.$$

In order to find an approximation to the exact solution of the variational problem (9), we attempt to represent the solution by an element from a given finite dimensional subspace $V_h \subset V$. This is known as the *Galerkin* method (*Bubnov-Galerkin* or *Ritz-Galerkin* method in the self-adjoint case) [18]. For a given finite dimensional subspace $V_h \subseteq V$ the variational form of the eigenvalue problem (9) then is the *discretized eigenvalue problem*:

Determine $\lambda_h \in \mathbb{R}$ and $u_h \in V_h$ such that

$$(10) \quad a(u_h, v_h) = \lambda_h b(u_h, v_h) \quad \text{for all } v_h \in V_h.$$

Since A is symmetric positive definite, it follows that $a(\cdot, \cdot)$ defines an inner product on V , and $b(\cdot, \cdot)$ is also an inner product on V .

There are many possible choices for the space V_h , see e.g., [11, 18]. For simplicity we discuss only the 2D case and let \mathcal{T}_h be a partition (triangulation) of the domain Ω into elements (triangles) T , such that

$$\bigcup_{T \in \mathcal{T}_h} T = \bar{\Omega},$$

any two distinct elements in \mathcal{T}_h share at most a common edge or a common vertex. For each element $T \in \mathcal{T}_h$ by $\mathcal{E}(T)$ and $\mathcal{N}(T)$ we denote the set of corresponding edges and vertices, respectively, and \mathcal{E}_h and \mathcal{N}_h denotes all edges and vertices in \mathcal{T}_h . Likewise, we define h_T as the *diameter* (the length of the longest edge) of an element. For each edge E we denote its length by h_E and the unit normal vector by \vec{n}_E . We set $h := \max_{T \in \mathcal{T}_h} h_T$. We say that the triangulation is *regular*, see [18], if there exists a positive constant ρ such that

$$\frac{h_T}{d_T} < \rho,$$

with d_T being the diameter of the largest ball that may be inscribed in element T , i.e., the minimal angle of all triangles in \mathcal{T}_h is bounded away from zero.

Consider a regular triangulation \mathcal{T}_h of Ω and the set of polynomials \mathbb{P}_p of total degree $p \geq 1$ on \mathcal{T}_h , which vanish on the boundary of Ω , see, e.g., [11]. Then the Galerkin discretization of (10) with $V_h^p \subset V$, $\dim V_h^p = n_h$, chosen as

$$V_h^p(\Omega) := \{v_h \in C^0(\bar{\Omega}) : v_h|_T \in \mathbb{P}_p \text{ for all } T \in \mathcal{T}_h \text{ and } v_h = 0 \text{ on } \partial\Omega\},$$

is called *finite element discretization*. The Finite Element Method (FEM) [18] is a Galerkin method where V_h is the subspace of piecewise polynomial functions, i.e., functions that are continuous in Ω and that are polynomial on each $T \in \mathcal{T}_h$. To simplify the presentation, here we only consider \mathbb{P}_1 finite elements, i.e., $p = 1$, and use $V_h := V_h^1$. The motivation to use piecewise polynomials is that in this case the computational work in generating the system matrices is small and they are sparse, since the space V_h then has a canonical basis of functions with small support. The basis $\{\varphi_h^{(1)}, \dots, \varphi_h^{(n_h)}\}$ is then a *Lagrange* or *nodal basis* [18] and an eigenfunction u_h is determined by its values at the n_h grid points of \mathcal{T}_h and it can be written as

$$u_h = \sum_{i=1}^{n_h} u_{h,i} \varphi_h^{(i)}$$

and the discretized problem (10) reduces to a *generalized algebraic eigenvalue problem* of the form

$$(11) \quad K_h \mathbf{u}_h = \lambda_h M_h \mathbf{u}_h,$$

where the matrices

$$K_h := [a(\varphi_h^{(j)}, \varphi_h^{(i)})]_{1 \leq i, j \leq n_h}, \quad M_h := [b(\varphi_h^{(j)}, \varphi_h^{(i)})]_{1 \leq i, j \leq n_h}$$

are called *stiffness* and *mass* matrix, respectively. The coordinate vector \mathbf{u}_h associated with the eigenfunction u_h is defined as

$$\mathbf{u}_h := [u_{h,i}]_{1 \leq i \leq n_h}.$$

Since $a(\cdot, \cdot)$ and $b(\cdot, \cdot)$ are bounded and symmetric and associated with scalar products, the resulting matrices K_h, M_h are symmetric and positive definite. Thus, see e.g. [57, §15.3], for the discrete eigenvalue problem (11) there is a full set of real M_h -orthogonal eigenvectors given by the columns of a matrix $\mathbf{U}_h \in \mathbb{R}^{n_h, n_h}$, and all eigenvalues $\lambda_h^{(i)}$, $i = 1, 2, \dots, n_h$, are real and positive, i.e., we have

$$\mathbf{U}_h^T K_h \mathbf{U}_h = \text{diag}(\lambda_h^{(1)}, \dots, \lambda_h^{(n_h)}), \quad \mathbf{U}_h^T M_h \mathbf{U}_h = I_n.$$

Furthermore, it is well-known, e.g. [4, Equation (8.42), p. 699], [66, equation (23), p. 223] or [19, 70], that for *conforming approximations*, i.e., if $V_h \subset V$, then the *Courant-Fischer min-max characterization* implies that the exact eigenvalues are approximated from above, i.e.,

$$\lambda^{(i)} \leq \lambda_h^{(i)}, \quad i = 1, 2, \dots, n_h.$$

For the comparison of the pros and cons in AFEM and AMLS we restrict ourselves to the computation of a few of the smallest eigenvalues and their corresponding eigenvectors/eigenfunctions.

3. Automated Multilevel Substructuring

In this section we briefly review the Automated Multilevel Substructuring method (AMLS) [9, 42], including some small improvements of its original formulation. For simplicity of presentation we consider a problem of the form (11) which either arises from discrete FE modeling or from the FE discretization of a 2D elliptic PDE problem. The AMLS method needs as an input just the two matrices K_h and M_h and a user-supplied cutoff value $\lambda_c > 0$ and then works purely algebraically to determine all approximate eigenvalue/eigenvector pairs $(\lambda_h^{(i)}, \mathbf{u}_h^{(i)})$ associated with eigenvalues $0 < \lambda_h^{(i)} \leq \lambda_c$. The AMLS method can be viewed as an enhancement of the well-established *component mode synthesis method* of [60].

The first step in AMLS is to compute a nested dissection reordering [30], based on the computation of a set of vertex separators in the unweighted graph induced by the stiffness matrix and apply it to *both* stiffness and mass matrix. Formally this reordering yields a permutation matrix $P \in \mathbb{R}^{n_h, n_h}$, such that $K := P^T K_h P$, $M := P^T M_h P$ are block-structured, as illustrated in Figure 1. In the FE setting one obtains the same block structure in both matrices. Setting $w = P^T \mathbf{u}_h$, we then have the block-structured (sparse) generalized eigenvalue problem

$$(12) \quad K w = \lambda_h M w.$$

Note that in practice one does not apply a permutation matrix, but just relabels the indices in the eigenvectors and undoes this relabeling when the eigenvectors

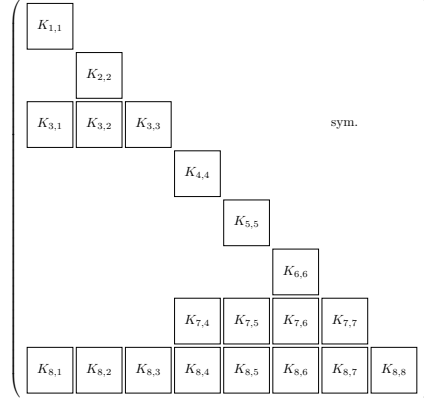


FIGURE 1. Symmetric stiffness matrix K with two levels of nested dissection reordering. The blocks $K_{1,1}$, $K_{2,2}$, $K_{4,4}$, $K_{5,5}$, and $K_{6,6}$ are substructure blocks, the blocks $K_{3,3}$, $K_{7,7}$, $K_{8,8}$ are coupling blocks. The first level of substructuring consists of the blocks $K_{1:3,1:3}$, $K_{4:7,4:7}$, and $K_{8,8}$.

\mathbf{u}_h of the original problem are needed. The exact choice and number of partitions in the nested dissection reordering is irrelevant for the following description of the AMLS algorithm, but the structure is used heavily to improve the performance of the method when it is applied to a concrete eigenvalue problem. The resulting matrices are sparse block matrices with the diagonal blocks called *substructure blocks* and *coupling blocks*. Coupling blocks correspond to the vertex separators in the graph described by the stiffness matrix, whereas the substructure blocks correspond to substructures in the graph. Since the mass and stiffness matrices arise from the assembly of the inner products between the locally (via the mesh structure) defined basis functions, the resulting block structure partitioning can be viewed as a subdivision of the continuous domain [9]. However, since the method works in a completely algebraic fashion, it can also be applied if the matrices K, M are not associated with any finite element model if the graph partitioning software is applied to the combined graph of both matrices.

The second step in AMLS is to compute a block Cholesky decomposition [21, 27] $K = LDL^T$ of the reordered stiffness matrix, so that L is a block lower triangular and D is a block diagonal matrix. Since K is positive definite, this block Cholesky decomposition exists without employing any pivoting strategy.

REMARK 3.1. The second step of AMLS may be infeasible when K or some of the diagonal blocks are singular or close to being singular. Moreover, since this is a direct factorization there will be a non-negligible fill-in which may necessitate out-of-core algorithms in the implementation. It is possible to also include singular $K_{j,j}$ matrices and then using pivoting move singular blocks of K to the bottom of the matrix. This will, however, partially destroy the sparsity and increase the size of the final block. For the comparison of the two methods we restrict ourselves to the case of invertible $K_{j,j}$.

Using the block Cholesky factors, another change of basis is performed. Setting $M_L := L^{-1}ML^{-T}$, $w_L := L^T w$ yields the transformed problem

$$(13) \quad K_L w_L = \lambda_h M_L w_L,$$

with block diagonal matrix $K_L = \text{diag}(K_{1,1}, \dots, K_{\ell,\ell})$ and blocks $K_{j,j}$, $j = 1, \dots, \ell$, of sizes n_1, \dots, n_ℓ . Due to the structure of L, L^{-1} , it follows that $M_L = [M_{i,j}]$ retains the nested dissection reordering block structure if it is partitioned analogously. Furthermore, since the transformation is a congruence transformation, K_L and M_L are still symmetric positive definite due to Sylvester's law of inertia [27, Theorem 8.1.17].

In the third step of the AMLS algorithm one solves the local eigenvalue problems

$$K_{j,j} \tilde{w}_j = \tilde{\lambda}_j M_{j,j} \tilde{w}_j, \quad j = 1, \dots, \ell,$$

associated with the diagonal blocks of K_L, M_L , using any of the well-established methods for small dense eigenvalue problems [3, 10]. Let $\tilde{\Lambda} = \text{diag}(\tilde{\Lambda}_1, \dots, \tilde{\Lambda}_\ell)$ be the block matrix with diagonal blocks containing all the computed eigenvalues $\tilde{\lambda}_j^{(k)}$, $j = 1, \dots, \ell$, $k = 1, \dots, n_j$, of these subproblems, and let $\tilde{W} = \text{diag}(\tilde{W}_1, \dots, \tilde{W}_\ell) \in \mathbb{R}^{n_h, n_h}$ be the corresponding block diagonal matrix of computed eigenvectors, i.e., $\tilde{W}_j = [w_j^{(1)}, \dots, w_j^{(n_j)}]$, $j = 1, \dots, \ell$.

The fourth step in the AMLS method is to perform a *modal truncation* in the sub-blocks, i.e., with the given cutoff value λ_c , we select the subset of eigenvalues in the blocks $\tilde{\Lambda}_j$ that fall below the cutoff λ_c , and set $\hat{\Lambda}_j$, $j = 1, \dots, \ell$, as the corresponding submatrices of $\tilde{\Lambda}_j$. Moreover, we define $\hat{\Lambda} = \text{diag}(\hat{\Lambda}_1, \dots, \hat{\Lambda}_\ell)$. Analogously, we define the block diagonal matrix $\hat{W} = \text{diag}(\hat{W}_1, \dots, \hat{W}_\ell)$, where each block \hat{W}_j contains only those eigenvectors from \tilde{W}_j which correspond to the eigenvalues in $\hat{\Lambda}_j$.

REMARK 3.2. Note that here we use the fact that the smallest values of $\tilde{\lambda}_j^{(k)}$ are reasonable approximations of the smallest exact eigenvalues λ_h of (12), which follows from the Cauchy interlacing theorem, [27]. For more general problems this may not be justified.

The fifth step in the AMLS method is to project the matrix pencil (K_L, M_L) onto the space spanned by the columns of \hat{W} . Let \hat{W}^\dagger be the Moore-Penrose pseudo-inverse of \hat{W} [27, p. 290], then we consider the projected eigenvalue problem

$$(14) \quad (\hat{W}^T K_L \hat{W}) \hat{W}^\dagger w_L = \lambda_S (\hat{W}^T M_L \hat{W}) \hat{W}^\dagger w_L,$$

and we set $K_S := \hat{W}^T K_L \hat{W}$, $M_S := \hat{W}^T M_L \hat{W}$, $w_S := \hat{W}^\dagger w_L$. Then the projected eigenvalue problem (14) has the form $K_S w_S = \lambda_S M_S w_S$. Note that the Moore-Penrose pseudo-inverse \hat{W}^\dagger does not have to be computed explicitly if we want to determine the eigenvectors of (13), since $w_L = \hat{W} w_S$. Note further, that we can exchange the fourth and the fifth step in the AMLS method. Let Z be a matrix containing a subset of the columns of the identity matrix, then we can choose it such that

$$\hat{W} = \tilde{W} Z \quad \text{and} \quad \hat{\Lambda} = Z^T \tilde{\Lambda} Z.$$

Accordingly, $K_S = Z^T \tilde{W}^T K_L \tilde{W} Z$ and we reversed the order of steps four and five. Moreover, notice that $K_S = \hat{\Lambda}$.

The sixth step of the AMLS algorithm is to solve the projected eigenvalue problem (14), again with appropriate solvers from [3, 10] and to use the obtained m smallest eigenvalues $\tilde{\lambda}_S^{(i)}, i = 1, \dots, m$, as approximations to the smallest m eigenvalues $\lambda_h^{(i)}, i = 1, \dots, m$, of the problem (12). The corresponding eigenvector approximations $\tilde{w}^{(i)}, i = 1, \dots, m$, of the exact eigenvectors $w^{(i)}$ of (12) are then obtained from

$$\tilde{w}^{(i)} = L^{-T} \widehat{W} \tilde{w}_S^{(i)}, \quad i = 1, \dots, m.$$

If the FE matrices are of a recursive multilevel structure, then the described approach can be carried out in a multilevel fashion, i.e., if the discussed diagonal blocks K_{jj}, M_{jj} again have the structure of a FE matrix, which is typically the case, then we can apply the same idea recursively for several levels.

The AMLS method contains several tuning parameters that are typically chosen based on heuristics, e.g., in the choice of the cutoff criterion in step four, quite often, a tolerance θ is added, see [42], to select all eigenvalues $\tilde{\lambda}_j^{(k)}$ below $\theta\lambda_c$. The default value $\theta = 8.4^2$ is experimentally determined and we can confirm it as good choice in our experiments. One can justify this heuristic choice using the eigenvalue error bounds of [50]. Let us, for simplicity, consider a single level of substructuring and partition the transformed matrices K_S, M_S conformably as

$$K_S = \begin{bmatrix} K_S^{11} & \\ & K_S^{22} \end{bmatrix}, \quad M_S = \begin{bmatrix} M_S^{11} & M_S^{12} \\ M_S^{12T} & M_S^{22} \end{bmatrix},$$

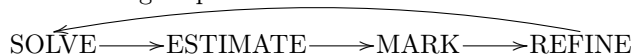
where M_S^{11} and M_S^{22} are identity matrices. Then K_S^{11}, M_S^{11} contain the substructure blocks, and K_S^{22}, M_S^{22} the coupling blocks. Theorem 2.5 in [50, p. 651] bounds the maximum difference between an exact eigenvalue $\lambda_h^{(i)}$ and an eigenvalue approximation $\tilde{\lambda}_j^{(k)}$ that was computed by ignoring the off-diagonal blocks. If we assume that the modal truncation $\tilde{\lambda}_j^{(k)} \leq \theta\lambda_c$ found all eigenvalue approximations corresponding to exact eigenvalues $\lambda_h^{(i)} \leq \lambda_c$, then we are implicitly assuming that $\|M_S^{12}\|_2 \leq \frac{\theta-1}{\theta}$ which for $\theta = 8.4^2$ yields $\|M_S^{12}\|_2 \leq 0.9858$. Since M_S contains identity matrices on its block diagonal, there is only one level of substructuring, and M_S is symmetric positive definite, $\|M_S^{12}\|_2 < 1$ holds and hence the choice of θ fits well.

It should be noted further, that making an efficient use of the extracted substructures in the eigenvalue problem (12) is an important part of the practical implementation of an AMLS method, since this will significantly reduce the fill-in in the off-diagonal blocks of the block Cholesky factorization.

4. The Adaptive Finite Element Method (AFEM)

The standard finite element method proceeds from the selection of a mesh and basis to the computation of a solution. However, it is well-known that the overall accuracy of the numerical approximation is determined by several factors: the regularity of the solution (smoothness of the eigenfunctions), the approximation properties of the finite element spaces, i.e., the search and test space, the accuracy of the eigenvalue solver and its influence on the total error. The most efficient approximations of smooth functions can be obtained using large higher-order finite elements (p -FEM), where the local singularities, arising e.g. from re-entrant corners, interior or boundary layers, can be captured by small low-order elements

(*h*-FEM) [17]. Unfortunately, in real-world applications, these phenomena are typically not known a priori. Therefore, constructing an optimal finite dimensional space to improve the accuracy of the solution requires refining the mesh and (or) basis and performing the computations again. A more efficient procedure tries to decrease the mesh size (*h-adaptivity*) or (and) to increase the polynomial degree of the basis (*p-refinement*) automatically such that the accurate approximation can be obtained at a lower computational cost, retaining the overall efficiency. This adaptation is based on the local contributions of global *a posteriori error estimates*, the so-called *refinement indicators*, extracted from the numerical approximation. This algorithmic idea is called *Adaptive Finite Element Method (AFEM)* and can be described via the following loop



The number of manuscripts addressing adaptive finite element methods is constantly growing, and its importance cannot be underestimated. On the other hand mostly the publications do not deal with PDE eigenvalue problems but rather treat PDE boundary value problems. In the following sections we present a small fraction of material presented in [1, 4, 5, 6, 11, 13, 18, 23, 25, 26, 32, 31, 34, 35, 40, 44, 46, 47, 54, 55, 58, 59, 62, 63, 65, 66, 68].

The AFEM formulation that we will employ is based on the ansatz for the standard finite element method in Section 2. Since the AFEM will involve several levels of discretization with different mesh sizes h , we address this issue by slight modification of the notation introduced in Section 2. The label ℓ associated with the triangulation \mathcal{T}_ℓ indicates the refinement level of the mesh in the refinement hierarchy obtained by the adaptive FEM. We will assume that $\mathcal{T}_\ell \subset \mathcal{T}_{\ell+1}$, i.e., no coarsening is performed, and denote by N_ℓ and n_ℓ the maximal refinement level and the number of degrees of freedom associated with \mathcal{T}_ℓ , respectively. We will denote the finite dimensional space over the partition \mathcal{T}_ℓ as V_ℓ and the associated Galerkin approximation as (λ_ℓ, u_ℓ) . All other quantities are defined analogously as in Section 2 by the index h with ℓ .

The application of the adaptive FEM to the variationally stated eigenvalue problem (9) yields the following scheme: first the eigenvalue problem is solved on some initial mesh \mathcal{T}_0 to provide a finite element approximation (λ_ℓ, u_ℓ) of the continuous eigenpair (λ, u) . Afterwards, the total error in the computed solution is estimated by some *error estimator* η_ℓ .

If the estimate for the global error is sufficiently small, then the adaptive algorithm terminates and returns (λ_ℓ, u_ℓ) as a final approximation, otherwise, the local contributions of the error are estimated on each element. A local *error indicator* (*refinement indicator*) for an element $T \in \mathcal{T}_\ell$ is usually denoted by η_T and related to a global error estimator η_ℓ through

$$\eta_\ell = \left(\sum_{T \in \mathcal{T}_\ell} \eta_T^2 \right)^{1/2}.$$

Based on these estimators, the elements for refinement are selected and form the set $\mathcal{M}_\ell \subset \mathcal{T}_\ell$ of *marked elements*.

The process of selecting the elements of \mathcal{M}_ℓ is called the *marking strategy*, and typical heuristic choices based on numerical experiments are discussed in [12, 13, 20]. It should be noted that marking an element actually means marking all its edges. The refinement of the finite element space can be performed using various

techniques like moving grid points (*r-refinement*), subdividing elements of a fixed grid (*h-refinement*), applying locally higher-order basis functions (*p-refinement*) or any combinations of those [17]. For the sake of exposition, we discuss only the *h-refinement* of the elements, namely the longest-edge bisection [54, §4], and we do not discuss coarsening.

As we mentioned before, applying these refinement procedures may lead to nonconforming meshes with the so-called *hanging nodes*. Therefore, a *closure algorithm* [14] is applied to overcome this drawback and get a regular triangulation. For more details about adaptive refinement strategies see, e.g., [1, 17, 69].

In order to prove convergence of the AFEM one has to assume that the mesh refinement is done in such a way that a *saturation property* holds, as it has been proved for the Laplace eigenvalue problem in [15].

THEOREM 4.1 (Saturation property [15, Theorem 4.2]). *Let h_ℓ be the maximum mesh size on the ℓ -th mesh and let $\|\cdot\|_A := \sqrt{a(\cdot, \cdot)}$ denote the energy norm. Consider the adaptive FEM with sufficiently small maximal initial mesh size h_0 applied to the model problem (11). Then, there exists $0 \leq \rho < 1$ such that for all $\ell = 0, 1, \dots, n_\ell - 1$ the following inequalities hold:*

$$\begin{aligned} \|u - u_{\ell+1}\|_A^2 &\leq \rho \|u - u_\ell\|_A^2 + \lambda_{\ell+1}^3 h_\ell^4, \\ |\lambda - \lambda_{\ell+1}| &\leq \rho \|\lambda - \lambda_\ell\| + \lambda_{\ell+1}^3 h_\ell^4, \end{aligned}$$

where (λ, u) is an exact eigenpair of \mathcal{L} , λ_ℓ is an approximation of the eigenvalue λ on \mathcal{T}_ℓ and u_ℓ is the corresponding approximate eigenfunction.

With the saturation property given, a convergence proof for a specific AFEM in the computation of the smallest eigenvalues of (11) have been given in [15].

REMARK 4.2. Unfortunately, even the most accurate global error estimator itself does not guarantee the accuracy and efficiency of an adaptive algorithm. This can only be guaranteed if the desired eigenfunction has an approximate sparse representation in the used FE space, i.e., if they can be represented well by a small number of ansatz functions. This may, in particular, not be the case if several eigenfunctions are sought, which have singularities or oscillations in different regions of the spacial domain. Often the argument used in practice is that *nothing is better than a uniform mesh and brute force linear algebra* when studying eigenvalue problems. It should also be noted that almost all error estimators use inequalities that only hold up to an unknown constant and thus may lead to strong over- or underestimates of the true error. Finally, if the problem is sensitive to small perturbations (as it may be for non-self-adjoint non-normal problems), then even a very good and efficient error estimator may lead to large errors in eigenvalues and eigenfunctions. To include sensitivity of eigenvalues and eigenfunctions into the AFEM is currently a *very important open problem*.

The 'SOLVE step' in every step of the standard AFEM approach for eigenvalue problems (at least for reasonably fine levels) uses an iterative algebraic eigensolver, like an Arnoldi method [48]. Based on the computed eigenvalues/eigenvectors, then the a posteriori error estimates are determined and used to refine the grid. This approach, however, does not consider any influence of the errors in the algebraic eigensolver on the algorithm and most convergence or optimality results require that the eigenvalues and eigenvectors are computed exactly. Furthermore, since one may

have to solve many algebraic eigenvalue problems related to finer and finer grids and information from the previous steps of the adaptive procedure, like previously well approximated eigenvalues, is not used on the next level, computational costs for the algebraic eigenvalue problem often dominate the total computational cost.

A more efficient approach is an AFEM variant called AFEMLA [14, 15, 51], which incorporates the information obtained during the iterative solving of algebraic eigenvalue problems into the error estimation and the refinement process. Since the accuracy of the computed eigenvalue approximation cannot be better than the quality of the discretization, there is no need to solve the intermediate algebraic eigenvalue problems (that are used to compute the error estimates) up to very high precision if the discretization scheme guarantees only small precision. And even in the final step it is enough to solve the problem within an accuracy that fits to the discretized system. Also nested iterations, i.e., using actual eigenvector approximation as a starting vector for the eigenvalue computation on the refined grid, reduce the total cost significantly. AFEMLA therefore follows exactly *the idea of adaptive methods to achieve a desired accuracy with the minimal computational effort*.

An additional advantage of the AFEMLA algorithm over the standard AFEM is that it can be applied even without any knowledge of the underlying PDE problem and even if one is not able to construct an appropriate a posteriori error estimator. AFEMLA allows to construct an adaptive algorithm nevertheless, since it can be based on the algebraic residual, provided the problem is such that the residual information is sufficient to characterize the error. We are not addressing this problem here, however, also in this case, performing the subspace adaptation requires information about underlying meshes and matrices obtained at different discretization levels [51, 52, 53]

Let us now consider two consecutive partitions $\mathcal{T}_\ell \subset \mathcal{T}_{\ell+1}$ and associated finite element spaces $V_\ell \subset V_{\ell+1}$ with a finite element basis $\{\varphi_\ell^{(1)}, \dots, \varphi_\ell^{(n_\ell)}\}$ for V_ℓ and $\{\varphi_{\ell+1}^{(1)}, \dots, \varphi_{\ell+1}^{(n_{\ell+1})}\}$ for $V_{\ell+1}$. Let us assume for simplicity, that the mesh $\mathcal{T}_{\ell+1}$ is obtained by a uniform refinement of \mathcal{T}_ℓ .

With the Galerkin discretization followed by applying the Arnoldi process to the generalized eigenvalue problem

$$(15) \quad K_\ell \mathbf{u}_\ell = \lambda_\ell M_\ell \mathbf{u}_\ell$$

we get approximations $\tilde{\lambda}_\ell$ of the exact eigenvalues λ_ℓ on V_ℓ . With the approximation $\tilde{\mathbf{u}}_\ell$ to the corresponding eigenvector \mathbf{u}_ℓ , it follows that the corresponding approximate eigenfunction is given by

$$\tilde{u}_\ell = \sum_{i=1}^{n_\ell} \tilde{u}_{\ell,i} \varphi_\ell^{(i)},$$

where $\tilde{u}_{\ell,i}$ are the coefficients of the eigenvector $\tilde{\mathbf{u}}_\ell$, i.e., $\tilde{\mathbf{u}}_\ell := [\tilde{u}_{\ell,i}]_{i=1, \dots, n_\ell}$. We can compute the residual for this approximation and use this information for adaptation.

From a geometric point of view, it is our goal to enrich the space V_ℓ corresponding to the coarse mesh \mathcal{T}_ℓ by some further functions. Since V_ℓ is a subspace of $V_{\ell+1}$ corresponding to the mesh $\mathcal{T}_{\ell+1}$, every function from V_ℓ can be expressed

as a linear combination of functions from $V_{\ell+1}$. Thus,

$$\tilde{\mathbf{u}}_\ell = \sum_{i=1}^{n_\ell} \tilde{u}_{\ell,i} \varphi_\ell^{(i)} = \sum_{i=1}^{n_{\ell+1}} \hat{u}_{\ell+1,i} \varphi_{\ell+1}^{(i)},$$

with an appropriate coefficient vector $\hat{\mathbf{u}}_{\ell+1} = [\hat{u}_{\ell+1,i}]_{i=1,\dots,n_{\ell+1}}$. The relationship between coefficient vectors $\hat{\mathbf{u}}_{\ell+1}$ and $\tilde{\mathbf{u}}_\ell$ can be described by multiplication with an easily constructed prolongation matrix P_ℓ . Therefore, the corresponding prolonged coordinate vector in the fine space $V_{\ell+1}$ associated with the computed eigenvector \tilde{u}_ℓ is given as

$$\hat{\mathbf{u}}_{\ell+1} = P_\ell \tilde{\mathbf{u}}_\ell$$

Let us denote by $(\hat{\lambda}_{\ell+1}, \hat{\mathbf{u}}_{\ell+1})$ an approximate eigenpair obtained from the prolongation of the eigenvector $\tilde{\mathbf{u}}_\ell$ on the finite space $V_{\ell+1}$, where $\hat{\lambda}_{\ell+1}$ is a generalized Rayleigh quotient corresponding to $\hat{\mathbf{u}}_{\ell+1}$.

REMARK 4.3. If the algebraic eigenvalue problem could be solved exactly, then $\tilde{\lambda}_\ell$ and $\hat{\lambda}_{\ell+1}$ would be equal. But, since eigenvalues usually cannot be computed exactly and since we work in finite precision arithmetic, roundoff errors, although not discussed here, have to be taken into account as well as an early termination of the iteration and therefore it is important to distinguish these values.

Based on $(\hat{\lambda}_{\ell+1}, \hat{\mathbf{u}}_{\ell+1})$ we can compute the corresponding algebraic residual associated with the Galerkin discretization of the original problem on the fine mesh $\mathcal{T}_{\ell+1}$, i.e.,

$$(16) \quad \hat{\mathbf{r}}_{\ell+1} = K_{\ell+1} \hat{\mathbf{u}}_{\ell+1} - \hat{\lambda}_{\ell+1} M_{\ell+1} \hat{\mathbf{u}}_{\ell+1}.$$

This gives us a natural way of estimating the error in the computed eigenfunction using the coarse grid solution combined with the fine grid information, namely we can prolongate the already computed approximation $\tilde{\mathbf{u}}_\ell$ from V_ℓ to $V_{\ell+1}$. Then every entry in the residual vector $\hat{\mathbf{r}}_{\ell+1}$ in (16) corresponds to the appropriate basis function from the fine space. Furthermore, we know that if the i -th entry in the vector $\hat{\mathbf{r}}_{\ell+1}$ is large, then the i -th basis function has a large influence on the solution, namely its support should be further investigated [41]. All these basis functions with large entries in the vector $\hat{\mathbf{r}}_{\ell+1}$ together with all basis functions from the coarse space V_ℓ form a basis for the new refined space. The decision on whether an entry in the residual vector is small or large may again be based on different criteria, see [20]. When we identified the basis functions that should be added to enrich our trial space, we start the marking procedure. Since every FEM basis function is associated with a specific node in the mesh, enriching the space by new basis function means marking the edge corresponding to its node. In order to avoid hanging nodes or irregular triangulations, we again mark some additional edges using a closure algorithm, i.e., if edge is marked and is not a reference edge (the longest edge) of the element, then we add the reference edge to the set of marked edges. After that we can perform the actual refinement to obtain a new mesh which will be an initial mesh for the next loop of our adaptive algorithm.

For more details on the AFEMLA algorithm and, in particular, for the error estimates involving the algebraic error for elliptic self-adjoint eigenvalue problems, we refer to [51, 53].

5. Comparison of AMLS and AFEM

In this section we compare the discussed AMLS and AFEM methods. For this we consider the case, where the mass and stiffness matrices K_ℓ, M_ℓ resulting from the finest AFEM mesh \mathcal{T}_{N_ℓ} are given as input to the AMLS method. Let $\varphi_\ell^{(1)}, \varphi_\ell^{(2)}, \dots, \varphi_\ell^{(n_\ell)}$, $\ell \in \{0, 1, \dots, N_\ell\}$, denote the ansatz functions after the ℓ -th refinement. Moreover, since we do not perform a coarsening, we let

$$\varphi_\ell^{(i)} = \varphi_{\ell'}^{(i)}, \quad i = 1, 2, \dots, n_\ell, \quad \ell' > \ell.$$

We know that all eigenfunction approximations \tilde{u}_ℓ computed by the AFEM will be linear combinations of the ansatz functions $\varphi_\ell^{(i)}$, $i = 1, 2, \dots, n_\ell$, such that

$$\tilde{u}_\ell = \sum_{i=1}^{n_\ell} \tilde{u}_{\ell,i} \varphi_\ell^{(i)}, \quad \tilde{u}_{\ell,i} \in \mathbb{R}.$$

Since $\mathcal{T}_\ell \subset \mathcal{T}_{\ell+1}$, we may as well write \tilde{u}_ℓ , $\ell = 0, 1, \dots, N_{\ell-1}$, as

$$\tilde{u}_\ell = \sum_{i=1}^{n_\ell} \tilde{u}_{\ell,i} \varphi_{\ell'}^{(i)}, \quad \ell' > \ell.$$

Since we have assumed linear ansatz functions, there will be one degree of freedom in the algebraic problem for every basis function $\varphi_\ell^{(i)}$ and hence $K_\ell, M_\ell \in \mathbb{R}^{n_\ell, n_\ell}$. For simplicity, let us order the degrees of freedom such that the i -th algebraic variable belongs to the ansatz function $\varphi_\ell^{(i)}$. Because there is no coarsening, we can compute the matrices associated to the coarser grids by either removing (in the case of a hierarchical FE method) the last rows and columns of matrices K_ℓ and M_ℓ or (in a more general FE method) by appropriate linear combinations.

If we apply the AMLS method to compute a set of eigenpairs of the matrix pencil (K_{N_ℓ}, M_{N_ℓ}) , then as described before, this means the following steps:

- (1) the computation of a nested dissection reordering,
- (2) a congruence transformation using the block LDL^T decomposition,
- (3) the computation of the eigenpairs of the diagonal pencils $(K_{j,j}, M_{j,j})$,
- (4) modal truncation.

The first two steps are changes of basis in the algebraic problem, while the last two steps are the selection of a suitable linear combination of degrees of freedom such that the desired eigenspace is spanned. Since every degree of freedom corresponds to the coefficient of one basis function in the continuous problem, we can express aforementioned operations also in the continuous setting. Thus, the AFEM repeatedly refines the mesh only to assure that the algebraic eigensolver can discard certain linear combinations of ansatz functions.

To analyze the space of functions that is removed during the modal truncation and to analyze the relationship of AMLS and AFEM, we first have to keep in mind that the AMLS method does not have a direct access to the operator \mathcal{L} that was discretized and sometimes, as we discussed in the introduction, there even is no such operator \mathcal{L} . Nevertheless, we know that the ansatz functions $\varphi_\ell^{(i)}$ exist and if we were to increase the number of degrees of freedom to infinity, the mass and stiffness matrices would be the exact representation of an (unknown) differential operator \mathcal{L} associated with the limiting PDE, and for this operator AMLS is a method that computes approximate eigenvalues and eigenfunctions on a given mesh.

Within the AMLS method we repeatedly change bases in the coordinate vector spaces. If we assume that the 0-th level ansatz functions $\psi_0^{(i)} := \psi^{(i)}, i = 1, 2, \dots, n$ are either arising from some FE or AFEM discretization with \mathbb{P}_1 ansatz functions $\psi^{(1)}, \psi^{(2)}, \dots, \psi^{(n)}$ of the continuous problem associated with a PDE, then the algebraic changes of basis can be expressed via these ansatz functions. In the AFEM case we could start with some refinement level ℓ and set

$$\psi^{(i)} = \varphi_\ell^{(i)}, \quad i = 1, 2, \dots, n_\ell.$$

But any other way of constructing the initial basis would be appropriate as well and then lead to an i -th basis function $\psi_k^{(i)}$ in the k -th AMLS step. To see what is happening, assume for simplicity a single level of substructuring with one coupling block of size n_2 and one substructure block of size n_1 , such that $n = n_1 + n_2$. Let the corresponding index sets be denoted by $\mathcal{I}_1 := \{1, 2, \dots, n_1\}$ and $\mathcal{I}_2 := \{n_1 + 1, \dots, n\}$, respectively. Note that the AMLS method retains the nested dissection reordering computed in the first step. Thus, there is always one substructure block and one coupling block on the diagonal of the transformed matrices.

In the first step of the AMLS method a nested dissection reordering π is computed, such that

$$\psi_1^{(i)} = \psi_0^{(\pi(i))} = \psi^{(\pi(i))}.$$

Obviously, reordering the degrees of freedom in the discrete problem corresponds to a renumbering of the ansatz functions.

If in the second step of the AMLS method we denote the entries of the block Cholesky matrix L^T as $(L^T)_{ij}$, then

$$\psi_2^{(i)} = \sum_{j=1}^n (L^T)_{ij} \psi_1^{(j)}, \quad i = 1, 2, \dots, n,$$

because we perform congruence transformations $L^{-1}(K - \lambda_h M)L^{-T}L^T\psi_1 = 0$, and $\psi_2 := L^T\psi_1$. Due to the block structure, $\psi_2^{(i)} = \psi_1^{(i)}, i \in \mathcal{I}_2$, for all ansatz functions of the coupling block. For the basis functions corresponding to the substructure blocks, we have

$$\psi_2^{(i)} = \psi_1^{(i)} + \sum_{j \in \mathcal{I}_2} (L^T)_{ij} \psi_1^{(j)}, \quad i \in \mathcal{I}_1,$$

i.e., the basis functions $\psi_2^{(i)}$ corresponding to the substructure block after the LDL^T decomposition are linear combinations of the basis functions $\psi_1^{(i)}, i \in \mathcal{I}_1$, corresponding to the substructure blocks *and* the basis functions $\psi_1^{(i)}, i \in \mathcal{I}_2$ corresponding to the coupling blocks.

If one uses an eigenvalue projection method, then the computation of the eigenvectors of the blocks in the third AMLS step corresponds to a change of basis with the new basis being the set of block diagonal eigenvectors contained in the matrix \widetilde{W} . Since \widetilde{W} is block diagonal and partitioned conformably, we are computing linear combinations of the $\psi_2^{(i)}$ within their respective blocks:

$$\psi_3^{(i)} \in \text{span}\{\psi_2^{(i)} \mid i \in \mathcal{I}_k, k = 1, 2\}, \quad i = 1, 2, \dots, n.$$

The modal truncation within the AMLS method corresponds to the selection of transformed functions $\psi_3^{(i)}$:

$$\psi_4^{(i)} = \begin{cases} \psi_3^{(i)} & \lambda_i \leq \theta\lambda_c, \\ 0 & \text{otherwise.} \end{cases}$$

Thus whatever the initial set of ansatz function was, we have the following observations:

- Except for the LDL^T decomposition, the basis functions corresponding to one block do not appear in the linear combinations of the basis functions corresponding to the other blocks;
- the basis functions $\psi_\ell^{(i)}$, $\ell \geq 2$ corresponding to the substructuring blocks depend on basis functions $\psi_1^{(i)}$, $i \in \mathcal{I}_1$ corresponding to the coupling blocks;
- the basis functions $\psi_\ell^{(i)}$ corresponding to the coupling blocks depend only on the basis functions corresponding to the coupling blocks.

In any case, the AMLS method is computing linear combinations of the initial ansatz functions that approximate the desired eigenfunctions, by projecting into subspaces which are constructed using linear combinations spanned by global functions, which are still represented in the original basis. Thus all original fine basis functions participate in the representation and the final eigenfunctions are constructed using global functions in algebraically constructed substructures.

In contrast to this, the AFEM is selecting (by a projection with unit vectors), except for the very last step when the mesh refinement is terminated, local basis functions and a globalization is only done in the final eigenvalue/eigenvector computation.

From this point of view, one can say that AMLS starts with a fine mesh and uses globalization in substructures to reduce the system size, while AFEM starts with the same fine mesh but uses selected local basis functions all the way to the final step.

5.1. Analysis of AMLS and AFEM. Suppose that we compute and assemble mass and stiffness matrices using an FE software. Depending on the software, this is done with or without explicitly using an available PDE. Nevertheless, the analytical theory of the finite element method holds as long as the FE software uses conforming finite elements. Therefore, we can apply the standard convergence results, e.g. [66, Theorem 3.7], and for the case of an infinite number of degrees of freedom the solution u_h of the discretized problem will be identical to the solution u of the continuous problem. Moreover, the theorem of Necas [54, Theorem 2] guarantees that u_h will be the unique solution to some (unknown to us) variationally stated problem. Therefore, we can always apply the finite element convergence theory even if we do not know the underlying PDE.

But these statements are made from a purely mathematical point of view. If we were able to discretize problems with an arbitrary fine step size, we might as well replace all practical models in engineering, physics, and chemistry with a model based on the interaction of elementary particles and thereby avoid the modeling error inherent to macroscopic models.

In practice, there are some other aspects of the AFEM and the AMLS method that are more relevant. At first an AFEM implementation requires knowledge of the

differential operator \mathcal{L} . It needs to be able to construct a partition of the domain Ω , discretize a given PDE, solve the algebraic eigenvalue problems, estimate the errors, and refine the mesh. All of these components must be integrated and cannot work independently of each other, but then if the corresponding theory is available, the convergence of the method can be proved. Furthermore, AFEM works with local basis functions until the very last step and thus the matrices stay sparse. However, many levels of refinement may be needed to achieve a desired level of accuracy. This is particularly difficult if several eigenfunctions have to be computed.

On the other hand, the AMLS method requires only the knowledge of the matrix pencil (K, M) and the cutoff parameter λ_c . This considerably simplifies the implementation, since only knowledge of standard black box numerical linear algebra tools is needed. Moreover, the AMLS method does not require the underlying PDE model, which means that the AMLS method is more suitable for an arbitrary generalized eigenvalue problems as long as the corresponding matrices have the required properties, i.e., symmetry and positive (semi-)definiteness. If this is not the case, then the method can still be applied but becomes partially heuristic.

Both methods are fully automatic. The AFEM does not require any user intervention to decide about a good mesh structure or the polynomial degree of the ansatz functions in different subdomains, instead it will infer this information from the problem when needed. The AMLS method requires a matrix pencil as input and determines all necessary information from the input as well. The fully automatic approach, though it may not always be fast, makes eigenvalue computations also accessible to non-experienced users.

It remains *an open question* how to optimally combine the two methods to obtain the best of both worlds. Both methods are in their current form not suited for more complex eigenvalue problems like the second order problems with damping discussed in the introduction. In AFEM no theory on error estimates is available for complex multiple eigenvalues or Jordan blocks, so even if one would apply the method it would be strongly heuristic. The only practical extension with some kind of error control is to use a homotopy approach as introduced in [16], but even there only very limited partial results are available. AMLS could in principle be applied but with the cutoff value being replaced by some other selection criterion. But there is no theory that this will work in the case of complex eigenvalues or Jordan blocks. See also the experiments in Section 6.

6. Numerical Experiments

Since all existing AMLS implementations are closed source, in the remainder of the paper, we consider our own MATLAB implementation of the method. In the current version we can treat symmetric generalized eigenvalue problems of the form (12) with positive definite stiffness matrices and positive (semi-)definite mass matrices, thus we can also deal with problems having eigenvalues at or close to infinity. Moreover, we complement the AMLS method with a subspace iteration method (SIM) [8, 73] to turn the subspace generated by the AMLS method into a good approximation of the invariant subspace [73]. Our implementation does not require any additional assumptions on the matrices beyond symmetry and positive (semi-)definiteness, i.e., they do not need to be necessarily finite element matrices. The matrix reorderings are computed with METIS [43] which computes the

vertex separators such that there are exactly two substructures on every level of substructuring.

In this section we will illustrate our AMLS implementation with several numerical examples and present some interesting observations about the method. From now on, we will refer to the original AMLS method described by [42] as the *vanilla AMLS method* (vAMLS) and our implementation as the subspace improved AMLS (sAMLS). We also investigate possible modifications and improvements which allow to turn AMLS into a stand-alone eigenvalue solver.

6.1. A Quadratic Eigenvalue Problem. Consider the quadratic eigenvalue problem (QEP)

$$(17) \quad (\mu_h^2 M_h + \mu_h C_h + K_h) \mathbf{v}_h = 0.$$

M_h and K_h are real symmetric positive definite mass and stiffness matrices, C_h is a real symmetric positive semidefinite structural damping matrix. For a theoretical analysis of quadratic eigenvalue problems, see [67]. Many practitioners stress the ability of the AMLS method to generate from the undamped problem (17), i. e., $C_h = 0$, a subspace which is close to an invariant subspace associated with certain eigenvalues of the fully damped problem. As much as this statement is true for the case of proportional damping (Rayleigh damping), there is no mathematical reason why it should be true in the general case.

For this example and this example only, we will denote eigenpairs of the generalized eigenvalue problem (GEP) (11) with $(\lambda_h, \mathbf{u}_h)$ and the eigenpairs of (17) with (μ_h, \mathbf{v}_h) .

In the case of proportional damping, it is assumed that there exist real parameters $\alpha, \beta > 0$ such that

$$C_h = \alpha K_h + \beta M_h.$$

Then the eigenvectors \mathbf{u}_h of (11) are also eigenvectors of (17) and only the eigenvalues change, i. e., there exist eigenvalues $\mu_h \in \mathbb{C}$ such that

$$(\mu_h^2 M_h + \mu_h C_h + K_h) \mathbf{u}_h = 0.$$

Therefore, a good approximation to any invariant subspace of the GEP will also be a good approximation to an invariant subspace of the QEP. With general (real symmetric) damping matrices, this is not necessarily true and we will demonstrate this point using a real-life problem.

To assess the quality of an invariant subspace \mathcal{S} , we compute eigenpair approximations within \mathcal{S} and the error of each eigenpair with respect to the whole space \mathbb{R}^{n_h} . Then we can evaluate the quality of the space by applying statistical measures to the set of errors, e. g., we can calculate minimum, maximum, or arithmetic mean, etc.

Let

$$P(t) := t^2 M_h + t C_h + K_h.$$

The relative backward normwise error for a computed eigenpair (μ_h, \mathbf{v}_h) can be calculated, see e. g. [67, §4.2.1], via

$$\eta_P(\mu_h, \mathbf{v}_h) = \frac{\|P(\mu_h) \mathbf{v}_h\|_2}{\|\mathbf{v}_h\|_2 [|\mu_h|^2 \|M_h\|_2 + |\mu_h| \|C_h\|_2 + \|K_h\|_2]}.$$

It should be noted that an exact eigenpair of the GEP (11) solves the QEP $(\mu_h^2 M_h + K_h) \mathbf{u}_h = 0$ if $\mu_h = \pm i \sqrt{\lambda_h}$, where i is the imaginary unit and we will use this fact

	$\ \cdot\ _2$	$\kappa_2(\cdot)$
K_h	$5.9803 \cdot 10^7$	$2.8114 \cdot 10^6$
M_h	$7.0909 \cdot 10^{-5}$	714.20
C_h	6.8600	∞

TABLE 1. Euclidean norm $\|\cdot\|_2$ and condition number $\kappa_2(X) := \|X\|_2\|X^{-1}\|_2$ for a small brake model, $n_h = 4669$.

to evaluate eigenpairs of (11). Since we deal with relative errors, a small backward error has the same magnitude as the machine epsilon ε . The forward error can be computed by multiplying the backward error with the condition number and for an eigenvalue of (17), the condition number can be estimated via

$$\kappa_P(\mu_h, \mathbf{v}_h) = \frac{|\mu_h|^2 \|M_h\|_2 + |\mu_h| \|C_h\|_2 + \|K_h\|_2}{|\mu_h| |\mathbf{v}_h^T (2\mu_h M_h + C_h) \mathbf{v}_h|}.$$

If we consider an exact eigenpair $(\lambda_h, \mathbf{u}_h)$ of (11) as an approximate eigenpair for the QEP (17) (*with damping*) by setting

$$\tilde{\mu}_h := i\sqrt{\lambda_h}, \tilde{\mathbf{v}}_h := \mathbf{u}_h,$$

then, assuming that we can compute the required matrix 2-norms, we obtain

$$\eta_P(\tilde{\mu}_h, \tilde{\mathbf{v}}_h) = \frac{\|\tilde{\mu}_h C_h \tilde{\mathbf{v}}_h\|_2}{\|\tilde{\mathbf{v}}_h\|_2 [|\tilde{\mu}_h|^2 \|M_h\|_2 + |\tilde{\mu}_h| \|C_h\|_2 + \|K_h\|_2]}.$$

This illustrates that we can cause a large backward error if $\|C_h \tilde{\mathbf{v}}_h\|_2$ and $\|C_h\|_2$ are sufficiently large compared to $\|M_h\|_2$ and $\|K_h\|_2$, which confirms our claim that eigenspaces of (11) and (17) may be completely different.

Among other examples we consider as set of test matrices the Harwell-Boeing BCS structural engineering matrices which are a collection of real-world generalized eigenvalue problems [22]. For these problems, the stiffness matrix often has a norm that is several magnitudes larger than the norm of the mass matrix. If the damping matrix is small in norm compared to the stiffness matrix, then it does not have much influence on the backward error. To illustrate this, consider the FE model of an industrial disk brake with an external load from a brake pad [33]. Here, the matrices were obtained with the FEM software INTES PERMAS [39] and their properties are listed in Table 1. The damping matrix has only 342 columns and rows with nonzero entries and according to the MATLAB `rank` function, the damping matrix has numerical rank 50.

In the following we show how well different invariant subspaces calculated based on (11) approximate invariant subspaces of (17) using the brake model matrices. Specifically, we will compare

- the subspace \mathcal{S}_e spanned by the eigenvectors corresponding to the $k = 554$ smallest eigenvalues of (11),
- the subspace \mathcal{S}_s calculated by the AMLS implementation (sAMLS, dimension 352),
- and the subspace \mathcal{S}_v computed by vanilla AMLS (vAMLS, dimension 554).

The eigenvectors corresponding to the smallest eigenvalues of (11) were computed using the MATLAB function `eigs`. As a starting vector, `opts.v0` in `eigs`,

Space	Minimum	Maximum	Mean	Median
\mathcal{S}_e	$1.3 \cdot 10^{-17}$	$6.4 \cdot 10^{-16}$	$1.4 \cdot 10^{-16}$	$9.8 \cdot 10^{-17}$
\mathcal{S}_s	$1.5 \cdot 10^{-15}$	$3.1 \cdot 10^{-9}$	$1.2 \cdot 10^{-10}$	$3.3 \cdot 10^{-11}$
\mathcal{S}_v	$6.5 \cdot 10^{-7}$	$3.2 \cdot 10^{-2}$	$1.5 \cdot 10^{-3}$	$4.8 \cdot 10^{-4}$

TABLE 2. Relative normwise backward error with respect to the GEP (11) for eigenpairs computed on different subspaces.

we used in all cases the vector of all ones in order to have reproducible results. In both AMLS variants, we used four levels of substructuring, i. e., there are 16 substructure blocks, and the cutoff $\lambda_c = 10^{10}$. sAMLS was run with the default settings whereas for vAMLS we set $\theta = 2.1^2$ (this value is used during modal truncation). Note that the default value for θ is 8.4^2 for both AMLS variants. By changing θ for vAMLS, we wanted to reduce the excessive dimension of the vAMLS subspace when run with the default settings (3419). All computations were performed in double precision $\varepsilon = 2.2 \cdot 10^{-16}$. For every subspace, we will compare minimum, maximum, arithmetic mean, and median of the errors of all calculated eigenpairs within a given subspace.

REMARK 6.1. The size of the subspace returned by vanilla AMLS is determined during the modal truncation step. Here, every eigenvalue $\tilde{\lambda}_j, j = 1, \dots, \ell$, of every diagonal block is compared individually to the cutoff λ_c . Clearly, the worst possible case is that all comparisons have the same result because then either *all* or *no* vectors are selected. As an example, consider a matrix pencil, for which for all j_1, j_2, k_1, k_2 ,

$$\tilde{\lambda}_{j_1}^{(k_1)} = \tilde{\lambda}_{j_2}^{(k_2)},$$

i. e., all diagonal blocks have only one eigenvalue (with corresponding algebraic multiplicity) and for all diagonal blocks it is the same value.

The disk brake problem is close to such a worst-case matrix pencil since 3116 out of $n_h = 4669$ possible distinct eigenvalues of the diagonal blocks are in the interval $[10^{11}, 10^{12}]$. Note that the largest generalized eigenvalue of (11) in this case is $\lambda_{h,\max} \approx 2 \cdot 10^{13}$, and the smallest eigenvalue is $\lambda_{h,\min} \approx 2 \cdot 10^6$.

In Table 2, we display the backward errors of the different spaces with respect to the GEP (11), i. e., we measure how well each subspace approximates an invariant subspace of (11). As expected, \mathcal{S}_e is a very good approximation to an invariant subspace, the maximal relative backward error is comparable to the machine epsilon ε . The vAMLS subspace is a noticeably worse approximation to an invariant subspace than \mathcal{S}_e but the subspace iterations of sAMLS make a real difference here, and decrease the mean and median backward error compared to \mathcal{S}_v significantly. For completeness, we also list the relative forward errors for (11) in Table 3.

In Table 4, we display the relative normwise backward error with respect to (17). The presence of the damping matrix leads to large changes and increases the backward error by several orders of magnitude for \mathcal{S}_e and \mathcal{S}_s . For the minimum and maximum backward error, \mathcal{S}_e contains considerably better eigenpair approximations but on average (mean, median), there is a hardly any difference between \mathcal{S}_e and \mathcal{S}_s with respect to our quantitative measures. Interestingly, the damping has hardly an impact on the vAMLS space; consequently, the average backward

Space	Minimum	Maximum	Mean	Median
\mathcal{S}_e	$3.2 \cdot 10^{-14}$	$1.2 \cdot 10^{-10}$	$4.8 \cdot 10^{-13}$	$1.3 \cdot 10^{-13}$
\mathcal{S}_s	$8.1 \cdot 10^{-11}$	$1.1 \cdot 10^{-6}$	$1.6 \cdot 10^{-7}$	$8.3 \cdot 10^{-8}$
\mathcal{S}_v	$8.4 \cdot 10^{-2}$	$1.4 \cdot 10^0$	$5.2 \cdot 10^{-1}$	$4.5 \cdot 10^{-1}$

TABLE 3. Relative normwise forward error with respect to the GEP (11) for eigenpairs computed on different subspaces.

Space	Minimum	Maximum	Mean	Median
\mathcal{S}_e	$3.5 \cdot 10^{-15}$	$5.8 \cdot 10^{-3}$	$2.6 \cdot 10^{-4}$	$8.5 \cdot 10^{-5}$
\mathcal{S}_s	$9.2 \cdot 10^{-13}$	$7.7 \cdot 10^{-3}$	$2.7 \cdot 10^{-4}$	$7.0 \cdot 10^{-5}$
\mathcal{S}_v	$7.1 \cdot 10^{-7}$	$3.1 \cdot 10^{-2}$	$1.6 \cdot 10^{-3}$	$5.8 \cdot 10^{-4}$

TABLE 4. Relative normwise backward error with respect to the QEP (17) for eigenpairs computed on different subspaces.

Space	Minimum	Maximum	Mean	Median
\mathcal{S}_e	$4.2 \cdot 10^{-10}$	$4.4 \cdot 10^0$	$2.0 \cdot 10^{-1}$	$1.2 \cdot 10^{-1}$
\mathcal{S}_s	$1.1 \cdot 10^{-7}$	$4.7 \cdot 10^0$	$3.5 \cdot 10^{-1}$	$2.1 \cdot 10^{-1}$
\mathcal{S}_v	$7.7 \cdot 10^{-2}$	$6.9 \cdot 10^0$	$5.9 \cdot 10^{-1}$	$4.8 \cdot 10^{-1}$

TABLE 5. Relative normwise forward error with respect to the QEP (17) for eigenpairs computed on different subspaces.

error difference to the other spaces shrinks from several magnitudes to a factor less than 10. Note that the backward errors of \mathcal{S}_v are comparable for GEP and QEP.

Since we have used the invariant subspaces of (11) to find approximate invariant subspaces for (17), apparently a small backward error is not as important as other factors. To see whether we are least close to an exact solution, we also analyzed the forward error for (17), the results are shown in Table 5.

We see that on average, the spaces are hardly distinguishable using our measures. Especially surprising is the fact that the backward and the forward errors of \mathcal{S}_v have the same order of magnitude with respect to (11) and (17).

From a mathematical point of view, there is no reason why an invariant subspace for equation (11) should resemble an invariant subspace for (17) and in large scale problems it has been observed that indeed this may not be the case [33]. Nevertheless, the projection of quadratic eigenvalue problems via AMLS subspaces is a standard procedure in engineering practice. Surprisingly, in many cases, the quantitative differences between the three spaces vanish when computing errors with respect to (17). In practice, AMLS seems to be able to handle larger problems than a standard Arnoldi-eigensolver and in conjunction with our examples, one can see why AMLS subspaces are so popular for dimensional reduction in mechanical engineering applications.

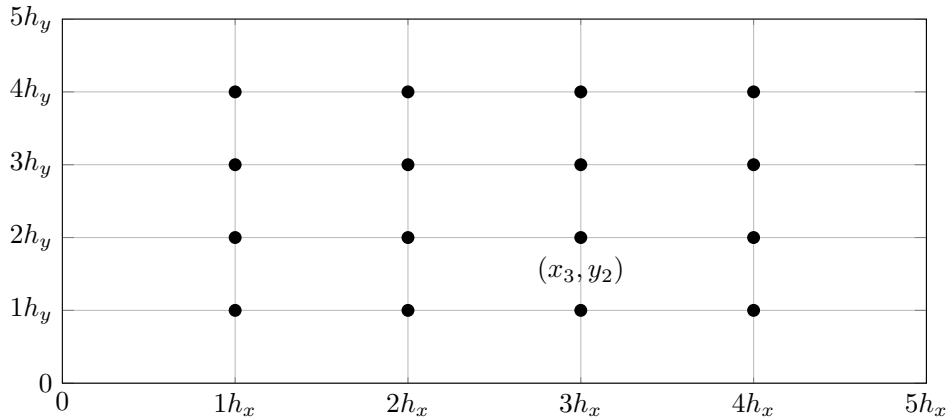


FIGURE 2. A grid over the rectangle $(0, 2) \times (0, 1)$ with $n = 4$ points on each axis. The inner nodes marked with black dots (\bullet) are the degrees of freedom of the discrete problem.

6.2. Isospectral Domains and AMLS. In this section we demonstrate a difficulty of the AMLS approach when dealing with isospectral domains. As a model problem we will use the simple Laplace eigenvalue problem (8) defined over the rectangular domain $\Omega = (0, \alpha) \times (0, \beta)$. For this choice the exact eigenpairs are known [19, Chapter VI, §4.1] and explicitly given as

$$(\lambda_{i,j}, u_{i,j}) = \left(\left(\frac{i^2}{\alpha^2} + \frac{j^2}{\beta^2} \right) \pi^2, \frac{2}{\sqrt{\alpha\beta}} \sin \frac{i\pi x}{\alpha} \sin \frac{j\pi y}{\beta} \right), \quad i, j = 1, 2, \dots$$

Obviously, the problems with domains $(0, \alpha) \times (0, \beta)$ and $(0, \beta) \times (0, \alpha)$ possess the same set of eigenvalues.

Let us discretize the problem with axis parallel quadrilateral elements and piecewise linear ansatz functions over n inner nodes on each axis with the boundary points given by (x, y) with $x \in \{0, \alpha\}$ or $y \in \{0, \beta\}$ and step sizes $h_x := \alpha(n + 1)^{-1}$ and $h_y := \beta(n + 1)^{-1}$ in x - and y -direction, respectively. For this discretization the eigenvectors will be pointwise exact in the grid nodes. In Figure 2 we show the domain $(0, \alpha) \times (0, \beta)$ with $\alpha = 2, \beta = 1$ and the associated mesh.

For the following experiments we chose $n_h = 32, \lambda_c = 100, \theta = 8.4^2, \alpha = 1, \beta = 32$, and $\alpha = 32, \beta = 1$, respectively. The analyzed problem has $n_c = 91$ eigenvalues $\lambda_h \leq \lambda_c$. Let us now investigate the properties of the subspace generated by the vAMLS method. At first, we notice that the matrix substructures obtained with METIS [43] for both domains are the same. Since, in both cases, the sparsity pattern of the stiffness matrix is the same and METIS graph partitioner works with the *unweighted* induced graph, the fact of different directional scaling is not taken into account.

Figure 3 shows the cosines of the angles between the subspace generated by the vAMLS method and the set of exact eigenvectors. The eigenvectors are sorted according to their corresponding eigenvalues in ascending order. Ideally, the vAMLS method computes the subspace spanned by the first $n_c = 91$ eigenvectors, i.e., for $i = 1, 2, \dots, n_c$, the cosine of the angles is one and zero otherwise. As we can see in Figure 3, the vAMLS method does not compute the desired eigenspace in the

second case and moreover, it clearly does not recognize the isospectral property of the two domains. Although the presented example can be handled through a special discretization or directly by the AFEM, the purpose of our investigation was different. In engineering problems the most efficient algorithms strongly benefit from exploiting symmetries which allow to reduce significantly the overall computational complexity. Since the AMLS method is a commonly used approach in structural mechanics, we were interested how the AMLS method deals with present symmetries. Furthermore, independent of modeling, both algebraic problems have the same eigenvalues and this should be visible in the results.

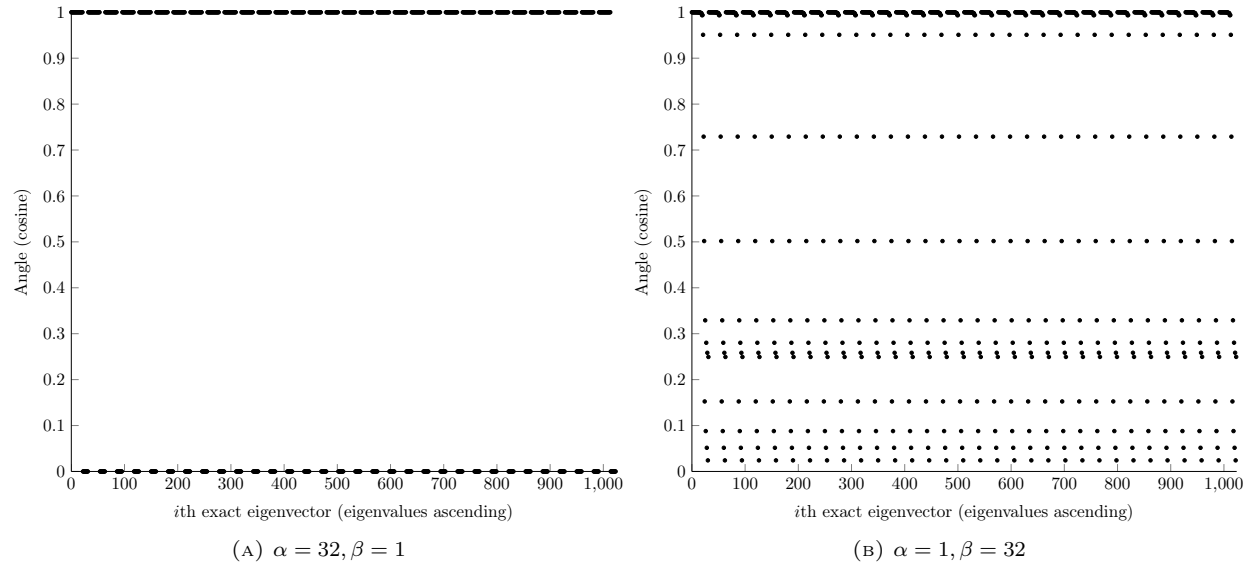


FIGURE 3. Subspace generated by vanilla AMLS with $n_h = 32, \lambda_c = 100$, and one level of substructuring with two substructure blocks.

7. Conclusion

We have discussed two of the most widely used methods for the solution of PDE eigenvalue problems as well as their advantages and disadvantages. The adaptive finite element method AFEM and its variant AFEMLA are provably efficient for standard elliptic problems. The automated multilevel substructuring method (AMLS) is widely used in practice for problems that require only approximate solutions. Since it does not compute a space with prescribed properties, its success is partially heuristic. In principle, both methods work with approximations of fine grid solutions. The difference is that in AFEM the approximation is an approximation on a coarse mesh that is obtained by leaving out nodes in the fine mesh and the associated basis functions, while in AMLS the reduction is achieved by choosing a basis of local eigenfunctions (which are represented by the fine mesh basis functions). Both methods work well on elliptic problems, but no theory is available for damped problems. AMLS is fast compared to standard eigensolvers and it is applicable to very large scale problems, and it often gives surprisingly good results even for damped problems. *However, what is needed is a major research effort to study AFEM or AMLS methods for practical PDE eigenvalue problems, with or without the given PDE model.*

References

- [1] M. Ainsworth and J. T. Oden, *A Posteriori Error Estimation in Finite Element Analysis*, Pure and Applied Mathematics, 2000, pp. xx+240.
- [2] A. Akay, Acoustics of friction, *Acoustical Society of America Journal*, vol. 111, no. 4, pp. 1525–1548, 2002.
- [3] E. Anderson, Z. Bai, C. Bischof, L. S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, eds., *LAPACK Users' Guide*, Software, Environments and Tools, 1999.
- [4] I. Babuška and J. E. Osborn, *Eigenvalue Problems. Handbook of Numerical Analysis Vol. II*, 1991, pp. 641–792.
- [5] I. Babuška and T. Strouboulis, *The Finite Element Method and Its Reliability*, Numerical Mathematics and Scientific Computation, 2001, pp. xii+802.
- [6] I. Babuška, J. Whiteman, and T. Strouboulis, *Finite Elements, An Introduction to the Method and Error Estimation*, 2010.
- [7] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst, *Templates for the Solution of Algebraic Eigenvalue Problems, A Practical Guide*, Software, Environments and Tools, 2000.
- [8] K.-J. Bathe, *Finite Element Procedures*, 1996.
- [9] J. K. Bennighof and R. B. Lehoucq, An Automated Multilevel Substructuring Method for Eigenspace Computation in Linear Elastodynamics, *SIAM J. Sci. Comput.* Vol. 25, no. 6, pp. 2084–2106, 2004.
- [10] L. S. Blackford, J. Choi, A. Cleary, E. D’Azevedo, J. Demmel, I. Dhillon, J. Dongarra, S. Hammarling, G. Henry, A. Petitdet, K. Stanley, D. Walker, and R. C. Whaley, *ScaLAPACK User's Guide*, Software, Environments and Tools, 1997.
- [11] D. Braess, *Finite Elements*, Third, Theory, fast solvers, and applications in elasticity theory, Translated from the German by Larry L. Schumaker, 2007, pp. xviii+365.

- [12] S. C. Brenner and C. Carstensen, “Finite Element Methods”, in: *Encyclopedia of Computational Mechanics, Vol. I*, ed. by E. Stein, R. de Borst, and T. J. R. Hugues, 2004, pp. 73–114.
- [13] S. C. Brenner and L. R. Scott, *The Mathematical Theory of Finite Element Methods*, Third, vol. 15, Texts in Applied Mathematics, 2008, pp. xviii+397.
- [14] C. Carstensen and J. Gedicke, An adaptive finite element eigenvalue solver of asymptotic quasi-optimal computational complexity, *SIAM J. Numer. Anal.* Vol. 50, no. 3, pp. 1029–1057, 2012.
- [15] C. Carstensen, J. Gedicke, V. Mehrmann, and A. Międlar, An adaptive finite element method with asymptotic saturation for eigenvalue problems, *Numer. Math.* Vol. 128, pp. 615–634, 2014.
- [16] C. Carstensen, J. Gedicke, V. Mehrmann, and A. Międlar, An adaptive homotopy approach for non-selfadjoint eigenvalue problems, *Numer. Math.* Vol. 119, no. 3, pp. 557–583, 2011.
- [17] Z. Chen, *Finite Element Methods and Their Applications*, Scientific Computation, 2005, pp. xiv+410.
- [18] P. G. Ciarlet, *The Finite Element Method for Elliptic Problems*, vol. 40, Classics in Applied Mathematics, Reprint of the 1978 original (North-Holland, Amsterdam), 2002, pp. xxviii+530.
- [19] R. Courant and D. Hilbert, *Methods of mathematical physics. Vol. I*, 1953, pp. xv+561.
- [20] W. Dörfler, A convergent adaptive algorithm for Poisson’s equation, *SIAM J. Numer. Anal.* Vol. 33, no. 3, pp. 1106–1124, 1996.
- [21] I. S. Duff, A. M. Erisman, and J. K. Reid, *Direct Methods for Sparse Matrices*, Monographs on Numerical Analysis, 1989.
- [22] Iain Duff, Roger Grimes, and John Lewis, Sparse Matrix Test Problems, *ACM Transactions on Mathematics Software*, vol. 15, no. 1, pp. 1–14, 1989.
- [23] H. Elman, D. Silvester, and A. Wathen, *Finite Elements and Fast Iterative Solvers with Applications in Incompressible Fluid Dynamics*, Second, Numerical Mathematics and Scientific Computation, 2014, pp. xiv+479.
- [24] K. Elssel and H. Voss, An a priori bound for automated multilevel substructuring, *SIAM J. Matrix Anal. Appl.* Vol. 28, no. 2, pp. 386–397, 2006.
- [25] A. A. Ern and J.-L. Guermond, *Theory and practice of finite elements*, vol. 159, Applied Mathematical Sciences, 2004, pp. xiv+524.
- [26] L. C. Evans, *Partial Differential Equations*, Second, vol. 19, Graduate Studies in Mathematics, 2010, pp. xxii+749.
- [27] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 4th ed., Dec. 2012.
- [28] D. Gallistl, Adaptive nonconforming finite element approximation of eigenvalue clusters, *Comput. Methods Appl. Math.* Vol. 14, no. 4, pp. 509–535, 2014.
- [29] D. Gallistl, An optimal adaptive FEM for eigenvalue clusters, *Numer. Math.* 2015.
- [30] A. George, Nested Dissection of a Regular Finite Element Mesh, *SIAM J. Numer. Anal.* Vol. 10, no. 2, pp. 345–363, 1973.
- [31] M. S. Gockenbach, *Partial differential equations*, Second, Analytical and numerical methods, 2011, pp. xx+654.
- [32] M. S. Gockenbach, *Understanding and Implementing the Finite Element Method*, 2006, pp. xvi+363.

- [33] N. Gräßner, S. Quraishi, C. Schröder, V. Mehrmann, and U. von Wagner, New Numerical Methods for the Complex Eigenvalue Analysis of Disk Brake Squeal, Submitted, 2014.
- [34] C. Grossmann and H.-G. Roos, *Numerical treatment of partial differential equations*, Universitext, Translated and revised from the 3rd (2005) German edition by Martin Stynes, 2007, pp. xii+591.
- [35] W. Hackbusch, *Elliptic Differential Equations*, vol. 18, Springer Series in Computational Mathematics, Translated from the author’s revision of the 1986 German original by Regine Fadiman and Patrick D. F. Ion, 1992, pp. xiv+311.
- [36] S. Hammarling, C. J. Munro, and F. Tisseur, An Algorithm for the Complete Solution of Quadratic Eigenvalue Problems, *ACM Transactions on Mathematical Software*, vol. 39, no. 3, 18:1–18:19, 2013.
- [37] U. L. Hetmaniuk and R. B. Lehoucq, “Multilevel methods for eigenspace computations in structural dynamics”, in: *Domain Decomposition Methods in Science and Engineering XVI*, vol. 55, Lect. Notes Comput. Sci. Eng. 2007, pp. 103–113.
- [38] V. Heuveline and R. Rannacher, A posteriori error control for finite element approximations of elliptic eigenvalue problems, *Adv. Comp. Math.* Vol. 15, no. 1–4, pp. 107–138, 2001.
- [39] Ingenieurgesellschaft für technische Software mbH, *PERMAS Production Description Version 15*, 2014.
- [40] C. Johnson, *Numerical Solution of Partial Differential Equations by the Finite Element Method*, Reprint of the 1987 edition, 2009, pp. ii+279.
- [41] C. Kamm, *A posteriori error estimation in numerical methods for solving self-adjoint eigenvalue problems*, MA thesis, Strasse des 17. Juni 136, 10623 Berlin: Technische Universität Berlin, Institut für Mathematik, 2007.
- [42] M. F. Kaplan, *Implementation of Automated Multilevel Substructuring for Frequency Response Analysis of Structures*, PhD thesis, Austin, TX, USA: University of Texas at Austin, 2001.
- [43] G. Karypis and V. Kumar, A Fast and Highly Quality Multilevel Scheme for Partitioning Irregular Graphs, *SIAM J. Sci. Comput.* Vol. 20, no. 1, pp. 359–392, 1998.
- [44] T. Kato, *Perturbation theory for linear operators*, Classics in Mathematics, Reprint of the 1980 edition, 1995, pp. xxii+619.
- [45] N. M. Kinkaid, O. M. O’Reilly, and P. Papadopoulos, Automotive disc brake squeal, *Journal of Sound and Vibration*, vol. 267, no. 1, pp. 105–166, 2003.
- [46] M. G. Larson and F. Bengzon, *The Finite Element Method: Theory, Implementation, and Applications*, vol. 10, Texts in Computational Science and Engineering, 2013, pp. xviii+385.
- [47] S. Larsson and V. Thomée, *Partial Differential Equations with Numerical Methods*, vol. 45, Texts in Applied Mathematics, 2003, pp. x+259.
- [48] R. B. Lehoucq, D. C. Sorensen, and C. Yang, *ARPACK User’s Guide, Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*, Software, Environments and Tools, 1998.
- [49] R.-C. Li, Y. Nakatsukasa, N. Truhar, and S. Xu, Erratum: Perturbation of Partitioned Hermitian Definite Generalized Eigenvalue Problems, *SIAM J. Matrix Anal. Appl.* Vol. 34, no. 1, Erratum for [50], pp. 280–281, 2013.

- [50] R.-C. Li, Y. Nakatsukasa, N. Truhar, and S. Xu, Perturbation of Partitioned Hermitian Definite Generalized Eigenvalue Problems, *SIAM J. Matrix Anal. Appl.* Vol. 32, no. 2, See also erratum [49], pp. 642–663, 2011.
- [51] V. Mehrmann and A. Międlar, Adaptive computation of smallest eigenvalues of self-adjoint elliptic partial differential equations, *Numer. Linear Algebra Appl.* Vol. 18, no. 3, pp. 387–409, 2011.
- [52] A. Międlar, *Functional perturbation results and the balanced AFEM algorithm for self-adjoint PDE eigenvalue problems*, Preprint 817, Berlin: DFG Research Center MATHEON, 2011.
- [53] A. Międlar, *Inexact Adaptive Finite Element Methods for Elliptic PDE Eigenvalue Problems*, PhD thesis, Technische Universität Berlin, Institut für Mathematik, 2011.
- [54] R. H. Nochetto, K. G. Siebert, and A. Veerer, “Theory of Adaptive Finite Element Methods: An Introduction”, in: *Multiscale, Nonlinear and Adaptive Approximation*, 2009, pp. 409–542.
- [55] R. H. Nochetto and A. Veerer, “Primer of adaptive finite element methods”, in: *Multiscale and adaptivity: modeling, numerics and applications*, vol. 2040, Lecture Notes in Math. 2012, pp. 125–225.
- [56] H. Ouyang, W. Nack, Y. Yuan, and F. Chen, Numerical analysis of automotive disc brake squeal: a review, *International Journal of Vehicle Noise and Vibration*, vol. 1, no. 3/4, pp. 207–231, 2005.
- [57] B. N. Parlett, *The Symmetric Eigenvalue Problem*, Classics in Applied Mathematics 20, 1998.
- [58] A. Quarteroni, *Numerical Models for Differential Problems*, Second, vol. 8, MS&A. Modeling, Simulation and Applications, Translated from the fifth (2012) Italian edition by Silvia Quarteroni, 2014, pp. xx+656.
- [59] A. Quarteroni and A. Valli, *Numerical Approximation of Partial Differential Equations*, Springer Series in Computational Mathematics, 2008.
- [60] Jr. R. R. Craig and M. C. C. Bampton, Coupling of Substructures for Dynamic Analyses, *AIAA Journal*, vol. 6, no. 7, pp. 1313–1319, 1968.
- [61] R. Rannacher, A. Westenberger, and W. Wollner, Adaptive finite element solution of eigenvalue problems: balancing of discretization and iteration error, *J. Numer. Math.* Vol. 18, no. 4, pp. 303–327, 2010.
- [62] P.-A. Raviart and J.-M. Thomas, *Introduction à l’Analyse Numérique des Équations aux Dérivées Partielles*, Collection Mathématiques Appliquées pour la Maîtrise, 1983, p. 224.
- [63] S. I. Repin, *A Posteriori Estimates for Partial Differential Equations*, vol. 4, Radon Series on Computational and Applied Mathematics, 2008, pp. xii+316.
- [64] J. Sethna, *Statistical Mechanics: Entropy, Order Parameters and Complexity*, 2006.
- [65] P. Šolín, *Partial Differential Equations and the Finite Element Method*, Pure and Applied Mathematics (New York), 2006, pp. xviii+472.
- [66] G. Strang and G. J. Fix, *An Analysis of the Finite Element Method*, Prentice-Hall Series in Automatic Computation, 1973, pp. xiv+306.
- [67] L. Taslamán, *Algorithms and Theory for Polynomial Eigenvalue problems*, PhD thesis, University of Manchester, School of Mathematics, 2014.
- [68] R. Verfürth, *A posteriori error estimation techniques for finite element methods*, Numerical Mathematics and Scientific Computation, 2013, pp. xx+393.

- [69] R. Verfürth, *A Review of A Posteriori Error Estimation and Adaptive Mesh-Refinement Techniques*, 1996.
- [70] H. F. Weinberger, *Variational methods for eigenvalue approximation*, Based on a series of lectures presented at the NSF-CBMS Regional Conference on Approximation of Eigenvalues of Differential Operators, Vanderbilt University, Nashville, Tenn., June 26–30, 1972, Conference Board of the Mathematical Sciences Regional Conference Series in Applied Mathematics, No. 15, 1974, pp. v+160.
- [71] C. Yang, W. Gao, Z. Bai, Xiaoye Z. S. Li, L.-Q. Lee, P. Husbands, and E. Ng, An Algebraic Substructuring Method for Large-Scale Eigenvalue Calculation, *SIAM J. Sci. Comput.* Vol. 27, no. 3, pp. 873–892, 2005.
- [72] Y. Yang, L. Sun, H. Bi, and H. Li, A note on the residual type a posteriori error estimates for finite element eigenpairs of nonsymmetric elliptic eigenvalue problems, *Appl. Numer. Math.* Vol. 82, pp. 51–67, 2014.
- [73] J. Yin, H. Voss, and P. Chen, Improving eigenpairs of automated multilevel substructuring with subspace iterations, *Computers and Structures*, vol. 119, pp. 115–124, 2013.