Enhanced algebraic substructuring for symmetric generalized eigenvalue problems

Vassilis Kalantzis and Lior Horesh

October 2022

EPrint ID: 2022.2

IBM Research
Thomas J. Watson Research Center

Preprints available from:
`https://researcher.watson.ibm.com/researcher/view.php?person=ibm-vkal`

Received <day> <Month>, <year>; | Revised <day> <Month>, <year>; | Accepted <day> <Month>, <year>

DOI: xxx/xxxx

ARTICLE TYPE

Enhanced algebraic substructuring for symmetric generalized eigenvalue problems

Vassilis Kalantzis | Lior Horesh

Thomas J. Watson Research Center
IBM Research
United States of America

Correspondence
Vassilis Kalantzis
Email: vkal@ibm.com

Present Address
1101 Kitchawan Road, Yorktown Heights
Westchester County, NY 10598

Abstract

This paper proposes a new substructuring algorithm to approximate the algebraically smallest eigenvalues and corresponding eigenvectors of a symmetric positive-definite matrix pencil (A, M). The proposed approach partitions the graph associated with (A, M) into a number of algebraic substructures and builds a Rayleigh-Ritz projection subspace by combining spectral information associated with the interior and interface variables of the algebraic domain. The subspace associated with interior variables is built by computing substructural eigenvectors and truncated Neumann series expansions of resolvent matrices. The subspace associated with interface variables is built by computing eigenvectors and associated leading derivatives of linearized spectral Schur complements. The proposed algorithm can take advantage of multilevel partitionings when the size of the pencil. Experiments performed on problems stemming from discretizations of model problems showcase the efficiency of the proposed algorithm and verify that adding eigenvector derivatives can enhance the overall accuracy of the approximate eigenpairs, especially those associated with eigenvalues located near the origin.

KEYWORDS:
Symmetric generalized eigenvalue problems; spectral Schur complements; algebraic substructuring; Neumann series; Taylor series

1 | INTRODUCTION

This paper considers the approximation of the $n_{ev} \in \mathbb{N}$ smallest eigenvalues and associated eigenvectors (eigenpairs) of symmetric generalized eigenvalue problems $Ax = \lambda Mx$, where the $n \times n$ matrices A and M are sparse and symmetric, and M is symmetric positive-definite (SPD). Eigenvalue problems of this form appear in several problems in science and engineering such as the dynamic analysis of large Finite Element models[1,2,3,4,5,6,7], spectral graph clustering[8], and electronic structure calculations[9].

The standard approach to compute partial spectral factorizations of symmetric matrix pencils is to perform a Rayleigh-Ritz projection onto a low-dimensional subspace which captures a good approximation of the invariant subspace associated with the n_{ev} smallest eigenvalues[10]. Perhaps the most popular approach to build this subspace is by means of a Krylov subspace algorithm, where the projection subspace is built in an iterative fashion and the Rayleigh-Ritz projection of the pencil (A, M) is carried out implicitly[11,12]. For symmetric eigenvalue problems, the standard Krylov subspace technique used is a restarted variant of the Lanczos iterative method[13,14,15], often combined with a shift-and-invert transformation[16]. While Krylov subspace methods are exceptionally powerful, their application becomes increasingly impractical as the value of n_{ev} increases, due to the cost associated with maintaining an orthogonal basis of the Krylov subspace even when restarting is used. Additionally, the accuracy of the approximate eigenpairs returned by Krylov subspace approaches generally exceeds the requirements of several applications in engineering and science, e.g., Finite Element modelling and analysis.

An approach that can take advantage of lower accuracy requirements in order to develop faster algorithms is that of *algebraic substructuring*. The main idea behind substructuring is to reduce the dimensionality of the eigenvalue problem by decomposing it into several non-overlapping substructures which can be handled in parallel. In particular, each substructure can be embedded in the subspace formed by its dominant eigenmodes, which gives rise to the Component Mode Synthesis method (CMS)[17,18,19,20,21]. In linear algebraic terms, CMS applies block Gaussian elimination to transform the eigenvalue problem associated with the pencil (A, M) into its Craig-Bampton form[22], followed by a computation of a number of eigenmodes from each substructure, e.g., those associated with eigenvalues below a cut-off threshold. From a Partial Differential Equation perspective, algebraic substructuring does not use any information regarding the physical geometry of the computational domain. Instead, interface boundaries and connections between adjacent substructures are defined algebraically, and the graph associated with the pencil (A, M) is partitioned into a number of algebraic substructures by a graph partitioner such as METIS[23].

The Automated MultiLevel Substructuring (AMLS) algorithm[24,25,26,27,28,29,5] is a multilevel extension of CMS in which the interface variables form a separate substructure and the graph associated with the pencil (A, M) is partitioned recursively into smaller substructures using nested dissection[30]. Once the leaf substructures are reached, AMLS computes a number of eigenmodes from each substructure (e.g., those below a cut-off threshold) and traverses the elimination tree in an upwards fashion, each time multiplying the interface eigenmodes at the current level with the corresponding block Gaussian elimination matrix. Due to its ability to quickly achieve dramatic reductions in the size of Finite Element models, AMLS has been shown to be quite efficient in frequency response and eigenvalue analysis of real-world engineering problems[31,32,7,33,34,35]. Nonetheless, the accuracy of the approximate eigenpairs returned by AMLS can be rather low for general algebraic eigenvalue problems. Moreover, as AMSLS is based on nested dissection, its suitability for execution on distributed memory computing environments is limited.

The algorithm proposed in this paper shares similarities with substructuring algorithms such as AMLS and CMS but builds the Rayleigh-Ritz projection subspace by exploiting spectral Schur complements[24,36,37,38,39,40,41]. More specifically, we partition the adjacency graph associated with (A, M) into $p \in \mathbb{N}$ algebraic substructures, where the vertices of each substructure are classified either as interior, if they are connected only to vertices located in the same substructure, or interface, if they are also connected to vertices located in neighboring substructures. The projection subspace associated with interior variables is built locally in each substructure by computing substructural eigenvectors and truncating Neumann series expansions of resolvent matrices[36,42]. On the other hand, the projection subspace associated with interface variables is built by following the technique in[36], where it is suggested to exploit derivatives of eigenvectors associated with the smallest eigenvalues of a zeroth-order approximation of the spectral Schur complement. The algorithm developed in this paper extends the latter idea by computing the leading eigenvector derivatives of first-order approximations instead. A numerical procedure to approximate these eigenvector derivatives is outlined in the supplement. Experiments on a few model problems suggest that the proposed algorithm can outperform techniques based on computing solely eigenvectors of first-order linearizations of spectral Schur complements, e.g., AMLS. Moreover, the proposed algorithm can achieve similar accuracy to that obtained by computing eigenvectors of second-order lizearizations, but at a much lower computational cost. Extensions to recursive p-way partitionings are also considered. While not actively pursued in this paper, the proposed algorithm can execute efficiently on distributed memory computing environments due to partitioning by edge-separators.

1.1 | Notation and organization

Throughout the rest of this paper we denote the eigenpairs of the matrix pencil (A, M) by $\left(\lambda_i, x^{(i)}\right)$, $i = 1, \dots, n$. Without loss of generality, we assume that all eigenvalues are positive and ordered as $0 < \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$.[1] The eigenvalues of the pencil (A, M) will be collectively denoted by $\Lambda(A, M)$, and we denote by $\texttt{range}(K)$ and $\texttt{span}(v_1, \dots, v_k)$ the column space of matrix K and linear span of vectors $v_1, \dots, v_k$, respectively. Moreover, we set $K = \texttt{blkdiag}(K_1, ..., K_N)$ as the block diagonal matrix created by aligning the input matrices $K_1, \dots, K_N$, along its diagonal. Finally, the identity matrix of size n will be denoted by I_n while $\mathcal{F}(K)$ will denote the matrix stemming by setting each non-zero entry of K equal to one.

This paper is organized as follows. Section 2 discusses the concept of p-way edge-separators in algebraic substructuring. Section 3 presents background on algebraic substructuring eigenvalue solvers with p-way partitioners and discusses current

[1] Alternatively, if λ_1 is negative, we work with the pencil $(A + \sigma M, M)$ for some real scalar $\sigma < \lambda_1$.

limitations. Section 4 presents a new algorithm to compute the algebraically smallest eigenvalues and associated eigenvectors of symmetric matrix pencils. Section 5 presents numerical experiments which outline the performance of the proposed technique and provide guidance on how to attain best performance. Finally, section 6 presents our concluding remarks.

2 | GRAPH PARTITIONING AND ALGEBRAIC SUBSTRUCTURING

Let $\Omega := (\mathcal{V}, \mathcal{I})$ denote a graph with a set of vertices $\mathcal{V}$ and edges $\mathcal{I} := \{(\alpha, \beta) \mid (\alpha, \beta) \in \mathcal{V}^2 \; \& \; \alpha \neq \beta\}$. A *p-way edge-separator* of the graph Ω is defined as a set of edges $\mathcal{I}_s \subseteq \mathcal{I}$ whose removal from the edge set $\mathcal{I}$ divides the vertices of the graph Ω into $p \in \mathbb{N}$ non-overlapping sets $\mathcal{V}_1, \dots, \mathcal{V}_p$, such that the induced subgraphs (algebraic substructures) $\Omega_1 := (\mathcal{V}_1, \mathcal{I}_1), \dots, \Omega_p := (\mathcal{V}_p, \mathcal{I}_p)$, are disjoint. A graph can have many such partitions, and a good edge-separator is one for which the cardinalities $|\mathcal{V}_\ell|$, $\ell = 1, \dots, p$, are approximately constant, and, at the same time, the number of edges between vertices residing in different subgraphs is small. Figure 1 shows a p-way partitioning of a graph that models a 4×10 grid. The edge-separator $\mathcal{I}_s$ is denoted by dashed lines. The vertices of each induced subgraph Ω_ℓ, $\ell = 1, \dots, p$, can be classified either as *interface*, if the corresponding vertex is incident to an edge included in the edge separator $\mathcal{I}_s$, or *interior* otherwise.

Let now $\mathcal{F}(A)$ and $\mathcal{F}(M)$ denote the $n \times n$ matrices formed by setting the non-zero values of matrices A and M equal to one, respectively. We can then exploit graph partitioning to create algebraic substructures by applying a p-way edge-separator to the graph associated with the matrix $\mathcal{F}(A) + \mathcal{F}(M)$. Interface vertices denote unknowns which are coupled with equations from more than one substructures. In contrast, interior vertices denote unknowns which are coupled with equations only from the substructure they reside. If we reorder the unknowns/equations associated with interior variables before those associated with interface variables, and overwrite matrices A and M by their permutation, we obtain the following sparsity pattern:

$$A := \begin{array}{c} \\ d_1 \\ \vdots \\ d_p \\ s \end{array} \begin{array}{c} \begin{matrix} d_1 & \cdots & d_p & s \end{matrix} \\ \begin{pmatrix} B_1 & & & E_1 \\ & \ddots & & \vdots \\ & & B_p & E_p \\ E_1^T & \cdots & E_p^T & C \end{pmatrix} \end{array}, \qquad M := \begin{array}{c} \\ d_1 \\ \vdots \\ d_p \\ s \end{array} \begin{array}{c} \begin{matrix} d_1 & \cdots & d_p & s \end{matrix} \\ \begin{pmatrix} M_{B_1} & & & M_{E_1} \\ & \ddots & & \vdots \\ & & M_{B_p} & M_{E_p} \\ M_{E_1}^T & \cdots & M_{E_p}^T & M_C \end{pmatrix} \end{array}, \tag{1}$$

where $s = \sum_{\ell=1}^{p} s_\ell = n - \sum_{\ell=1}^{p} d_\ell = n - d$, and d_ℓ and s_ℓ denote the corresponding number of interior and interface variables of the ℓth substructure. The off-diagonal matrices have a special nonzero pattern of the form $E_\ell = \left[0_{d_\ell,\phi_\ell}, \widehat{E}_\ell, 0_{d_\ell,\nu_\ell}\right]$, $M_{E_\ell} = \left[0_{d_\ell,\phi_\ell}, \widehat{M}_{E_\ell}, 0_{d_\ell,\nu_\ell}\right]$, where $\phi_\ell = \sum_{k=1}^{k<\ell} s_k$, $\nu_\ell = \sum_{k>\ell}^{k=p} s_k$, and $0_{\chi,\psi}$ denotes the zero matrix of size $\chi \times \psi$. Figure 2 shows the sparsity pattern of matrix $\mathcal{F}(A) + \mathcal{F}(M)$ after partitioning the graph associated with a Finite Element discretization of the Laplace operator into $p = 2$ (left) and $p = 4$ (right) substructures.

3 | *P*-WAY ALGEBRAIC SUBSTRUCTURING EIGENVALUE SOLVERS

This section discusses the main framework of algebraic substructuring eigenvalue solvers using p-way graph partitioners. For simplicity, our discussion focuses on single-level partitionings.

Let matrices A and M shown in (1) be written in a compact 2×2 block form:

$$A = \begin{array}{c} \\ d \\ s \end{array} \begin{array}{c} \begin{matrix} d & s \end{matrix} \\ \begin{bmatrix} B & E \\ E^T & C \end{bmatrix} \end{array}, \qquad M = \begin{array}{c} \\ d \\ s \end{array} \begin{array}{c} \begin{matrix} d & s \end{matrix} \\ \begin{bmatrix} M_B & M_E \\ M_E^T & M_C \end{bmatrix} \end{array},$$

where $B = \texttt{blkdiag}(B_1, \dots, B_p)$, $M_B = \texttt{blkdiag}(M_{B_1}, \dots, M_{B_p})$, $E = [E_1^T, \dots, E_p^T]^T$, and $M_E = [M_{E_1}^T, \dots, M_{E_p}^T]^T$. Define now the $n \times n$ block-lower triangular matrix

$$U = \begin{bmatrix} I_d & \\ E^T B^{-1} & I_s \end{bmatrix}.$$

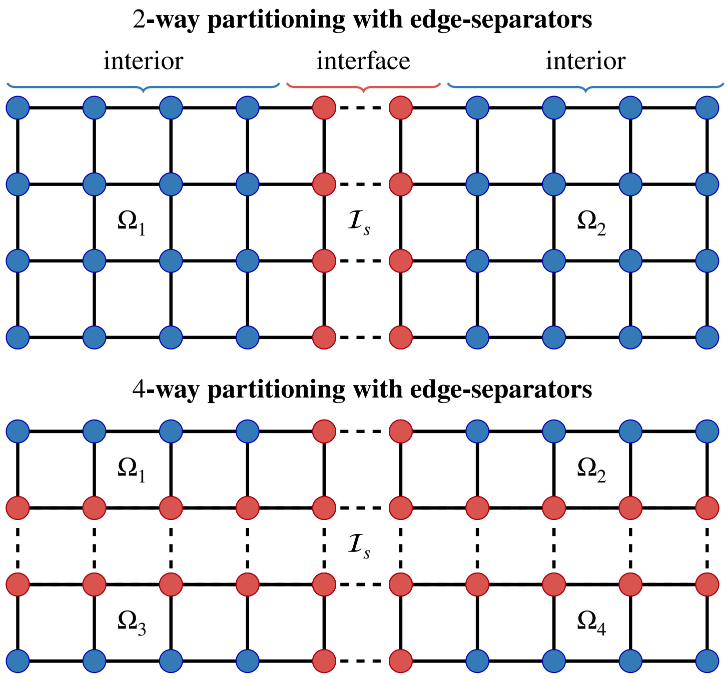

FIGURE 1 Illustration of a 4×10 grid partitioned into $p = 2$ (top) and $p = 4$ (bottom) substructures. Red filled circles denote interface variables while blue filled circles denote interior variables. Dashed lines form the edge-separator $\mathcal{I}_s$ and indicate couplings between interface variables residing in different substructures.

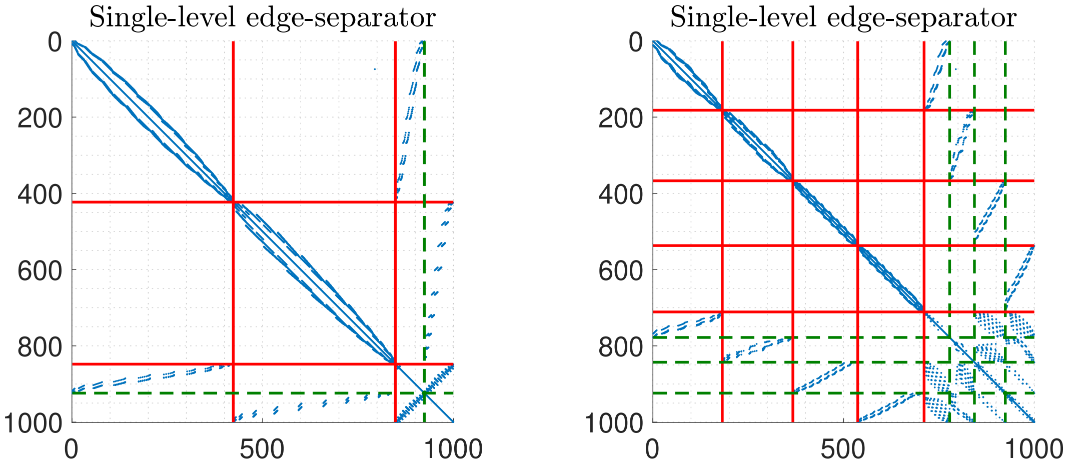

FIGURE 2 Sparsity pattern of matrix $\mathcal{F}(A) + \mathcal{F}(M)$ after partitioning with edge-separatos. Solid red lines indicate boundaries between interior variables. Dashed green lines indicate boundaries between interface variables located in neighboring subdomains. Left: $p = 2$. Right: $p = 4$.

Applying a congruence transformation with the matrix U^{-1} results in

$$\begin{aligned} U^{-1}AU^{-T}\left(U^T x\right) &= \lambda\, U^{-1}MU^{-T}\left(U^T x\right) \\ \begin{bmatrix} B & \\ & S \end{bmatrix} \hat{x} &= \lambda \begin{bmatrix} M_B & M_E - M_B B^{-1}E \\ M_E^T - E^T B^{-1} M_B & -M_S \end{bmatrix} \hat{x}, \end{aligned} \tag{2}$$

where $\hat{x} = U^T x$, and the $s \times s$ matrices S and M_S are equal to

$$S = C - E^T B^{-1} E,$$

and

$$M_S = -M_C - E^T B^{-1} M_B B^{-1} E + E^T B^{-1} M_E + M_E^T B^{-1} E.$$

The eigenvalues of the pencil (A, M) are identical to those of (2), while the eigenvectors x and $\widehat{x}$ are related by the equation $x = U^{-T}\widehat{x}$. The matrix $-M_S$ is SPD by virtue of being a principal submatrix of the SPD matrix $U^{-1}MU^{-T}$.

Definition 1. We will denote the eigenpairs of the pencils (B_ℓ, M_{B_ℓ}), $\ell = 1, \dots, p$, and $(S, -M_S)$, by $\left(\delta_i^{(\ell)}, v_i^{(\ell)}\right)$, $i = 1, \dots, d_\ell$, and $\left(\theta_j, y_j\right)$, $j = 1, \dots, s$, respectively, i.e.,

$$B_\ell v_i^{(\ell)} = \delta_i^{(\ell)} M_{B_\ell} v_i^{(\ell)} \quad \text{and} \quad S y_j = -\theta_j M_S y_j,$$

where we order $\delta_1^{(\ell)} \leq \cdots \leq \delta_{d_\ell}^{(\ell)}$, and $\theta_1 \leq \cdots \leq \theta_s$. The corresponding eigenvectors are scaled such that $\|v_i^{(\ell)}\|_{M_{B_\ell}} = \|y_j\|_{-M_S} = 1$, respectively, where for any SPD matrix K we define $\|x\|_K = \sqrt{x^T K x}$.

Let now $\kappa_\ell \in \mathbb{N}$ be an integer such that $1 \leq \kappa_\ell \leq d_\ell$ and define the matrices

$$V_{\kappa_\ell,\ell} = \left[v_1^{(\ell)}, \dots, v_{\kappa_\ell}^{(\ell)}\right], \qquad \Delta_{\kappa_\ell,\ell} = \mathtt{diag}\left(\delta_1^{(\ell)}, \dots, \delta_{\kappa_\ell}^{(\ell)}\right),$$

$$V = \mathtt{blkdiag}\left(V_{d_1,1}, \dots, V_{d_p,p}\right), \quad \Delta = \mathtt{blkdiag}\left(\Delta_{d_1,1}, \dots, \Delta_{d_p,p}\right),$$

and

$$Y = \left[y_1, \dots, y_s\right], \qquad \Theta = \mathtt{diag}\left(\theta_1, \dots, \theta_s\right).$$

Following the above notation, the eigenvector $\widehat{x}$ can be expressed as $\widehat{x} = \begin{bmatrix} V & \\ & Y \end{bmatrix} \begin{bmatrix} f^B \\ f^S \end{bmatrix}$, where the vectors $f^B \in \mathbb{R}^d$ and $f^S \in \mathbb{R}^s$ satisfy the eigenvalue equation

$$\begin{bmatrix} \Delta & \\ & \Theta \end{bmatrix} \begin{bmatrix} f^B \\ f^S \end{bmatrix} = \lambda \begin{bmatrix} I_d & V^T\left(M_E - M_B B^{-1} E\right) Y \\ Y^T\left(M_E - M_B B^{-1} E\right)^T V & I_s \end{bmatrix} \begin{bmatrix} f^B \\ f^S \end{bmatrix},$$

and the individual entries are equal to (e.g., see[43]):

$$\begin{aligned} f_i^B &= \frac{\lambda}{\delta_i - \lambda}\left(e_i^T V^T\left(M_E - M_B B^{-1} E\right) Y f^S\right), \; i = 1, \dots, d, \\ f_j^S &= \frac{\lambda}{\theta_j - \lambda}\left(e_j^T Y^T\left(M_E - M_B B^{-1} E\right)^T V f^B\right), \; j = 1, \dots, s. \end{aligned} \tag{3}$$

3.1 | The Rayleigh-Ritz perspective

The expressions in (3) suggest that when B is non-singular, the modulus of the entries f_i^B and f_j^S will be relatively large when $\lambda \approx \delta_i$ and $\lambda \approx \theta_j$, and relatively small otherwise. Since we are interested in computing the n_{ev} smallest eigenvalues of the pencil (A, M), the entries f_i^B and f_j^S should be relatively large when δ_i and θ_j are equal to the smallest eigenvalues of the matrix pencils (B_ℓ, M_{B_ℓ}), $\ell = 1, \dots, p$, and $(S, -M_S)$, respectively.

More specifically, let

$$V_{\kappa_B} = \mathtt{blkdiag}\left(V_{\kappa_1,1}, \dots, V_{\kappa_p,p}\right) \quad \text{and} \quad \Delta_{\kappa_B} = \mathtt{blkdiag}\left(\Delta_{\kappa_1,1}, \dots, \Delta_{\kappa_p,p}\right),$$

and define, for any $1 \leq \kappa_S \leq s$, the matrices

$$Y_{\kappa_S} = \left[y_1, \dots, y_{\kappa_S}\right] \quad \text{and} \quad \Theta_{\kappa_S} = \mathtt{blkdiag}\left(\theta_1, \dots, \theta_{\kappa_S}\right).$$

Ideally, the integers $\kappa_1, \dots, \kappa_p$, and κ_S satisfy $\lambda_{n_{ev}} \ll \mathtt{min}\left\{\mathtt{min}\left\{\delta_{\kappa_1+1}^{(1)}, \dots, \delta_{\kappa_p+1}^{(p)}\right\}, \theta_{\kappa_S+1}\right\}$. While $\lambda_{n_{ev}}$ is unknown, in several applications such as frequency response analysis, the eigenvalues $\lambda_1, \dots, \lambda_{n_{ev}}$, lie inside a real interval $[\alpha, \beta]$ which encapsulates exactly $n_{ev} \ll n$ eigenvalues. Thus, in practice, the above integers ideally satisfy $\beta \ll \mathtt{min}\left\{\mathtt{min}\left\{\delta_{\kappa_1+1}^{(1)}, \dots, \delta_{\kappa_p+1}^{(p)}\right\}, \theta_{\kappa_S+1}\right\}$. Following (3), the eigenvector $\widehat{x}$ can be approximated as

$$\widehat{x} \approx \begin{bmatrix} V_{\kappa_B} & \\ & Y_{\kappa_S} \end{bmatrix} \left[\widetilde{f}_1^B \cdots \widetilde{f}_{\kappa_B}^B \; \widetilde{f}_1^S \cdots \widetilde{f}_{\kappa_S}^S\right]^T,$$

where the real scalars $\widetilde{f}_1^B, \ldots, \widetilde{f}_{\kappa_B}^B$ and $\widetilde{f}_1^S, \ldots, \widetilde{f}_{\kappa_S}^S$, are provided by solving the eigenvalue problem

$$\begin{bmatrix} \Delta_{\kappa_B} & \\ & \Theta_{\kappa_S} \end{bmatrix} \widetilde{f} = \widetilde{\lambda} \begin{bmatrix} I_d & V_{\kappa_B}^T \left(M_E - M_B B^{-1} E\right) Y_{\kappa_S} \\ Y_{\kappa_S}^T \left(M_E - M_B B^{-1} E\right) V_{\kappa_B} & I_s \end{bmatrix} \widetilde{f}. \tag{4}$$

The eigenvalue problem listed in (4) is equivalent to a Rayleigh-Ritz projection of the pencil (A, M) onto the subspace

$$\mathcal{Z} = \mathtt{range}\left(\begin{bmatrix} I_d & -B^{-1}E \\ & I_s \end{bmatrix} \begin{bmatrix} V_{\kappa_B} & \\ & Y_{\kappa_S} \end{bmatrix} \right) = \mathtt{range}\left(\begin{bmatrix} V_{\kappa_B} & -(I_d - P_{\kappa_B})B^{-1}EY_{\kappa_S} \\ & Y_{\kappa_S} \end{bmatrix} \right),$$

where $P_{\kappa_B} = V_{\kappa_B} V_{\kappa_B}^T M_B$ denotes the M_B-orthogonal projector onto $\mathtt{range}\left(V_{\kappa_B}\right)$. The Rayleigh-Ritz eigenvalue $\widetilde{\lambda}_i$ is an approximation of the eigenvalue λ_i, $i = 1, \ldots, n_{ev}$, with a corresponding approximate eigenvector $\widetilde{x}^{(i)} = \begin{bmatrix} V_{\kappa_B} & -B^{-1}EY_{\kappa_S} \\ & Y_{\kappa_S} \end{bmatrix} \widetilde{f}^{(i)}$.

The computational cost of non-recursive algebraic substructuring is dominated by the cost associated with the partial spectral factorizations of the pencils (B_ℓ, M_{B_ℓ}), $\ell = 1, \ldots, p$, and $(S, -M_S)$. These eigenvalue problems are solved by the Implicitly Restarted Lanczos (IRL) with shift-and-invert[16,13,12,15]. On the other hand, the Rayleigh-Ritz eigenvalue problem (4) can be solved by the appropriate dense matrix routine in LAPACK/ScaLAPACK[44,45].

Algebraic substructuring with p-way partitioners is well-suited for distributed memory computing environments since the κ_ℓ, $\ell = 1, \ldots, p$, sought eigenpairs of the pencil (B_ℓ, M_{B_ℓ}) can be computed independently of each other. Additionally, the pencil $(S, -M_S)$ is naturally distributed by rows among the algebraic substructures due to partitioning by edge-separators. Thus, p can be set equal to the number of processing elements. Alternatively, it is possible to try different values of p and keep the one for which the matrix pencils $(B_1, M_{B_1}), \ldots, (B_p, M_{B_p})$, and $(S, -M_S)$ are roughly of the same size. This approach is geared towards sequential architectures and is our default strategy. Nonetheless, deriving an optimal strategy to choose p is rather challenging, since different values of p lead to different spectrums in the matrix pencils $(B_1, M_{B_1}), \ldots, (B_p, M_{B_p})$, and $(S, -M_S)$. Section 5 presents numerical experiments in which the number of partitions is varied.

4 | AN ALGEBRAIC SUBSTRUCTURING ALGORITHM

This section describes an algorithm based on algebraic substructuring. The main idea is to perform a Rayleigh-Ritz projection onto a subspace that can be written approximately as

$$\mathcal{Z} = \mathcal{Z}_0 + \mathcal{Z}_1 + \mathcal{Z}_2 + \ldots + \mathcal{Z}_{n_{ev}}, \tag{5}$$

where $\mathcal{Z}_0,\ \mathcal{Z}_i \in \mathbb{R}^n$, and $x^{(i)} \in \mathcal{Z}_0 + \mathcal{Z}_i$. The subspace $\mathcal{Z}_0$ includes information that is common for all sought eigenpairs $\left(\lambda_i, x^{(i)}\right)$, $i = 1, \ldots, n_{ev}$, while the subspace $\mathcal{Z}_i$ contains information that targets specifically the eigenpair $\left(\lambda_i, x^{(i)}\right)$. For the sake of simplicity, throughout the rest of this section we describe the proposed algorithm assuming that $\lambda_i \notin \Lambda(B, M_B)$, $1 \leq i \leq n_{ev}$.

4.1 | The eigenvector viewpoint

Let the eigenvector associated with the eigenvalue λ_i of the pencil (A, M) be partitioned as $x^{(i)} = \begin{bmatrix} u^{(i)} \\ y^{(i)} \end{bmatrix}$, where $u^{(i)} \in \mathbb{R}^d$ and $y^{(i)} \in \mathbb{R}^s$. The eigenvalue equation $(A - \lambda_i M)x^{(i)} = 0$ can be then written as $\begin{bmatrix} B - \lambda_i M_B & E - \lambda_i M_E \\ E^T - \lambda_i M_E^T & C - \lambda_i M_C \end{bmatrix} \begin{bmatrix} u^{(i)} \\ y^{(i)} \end{bmatrix} = 0$, from which it follows that

$$x^{(i)} = \begin{bmatrix} -(B - \lambda_i M_B)^{-1}(E - \lambda_i M_E)y^{(i)} \\ y^{(i)} \end{bmatrix}, \tag{6}$$

where $y^{(i)}$ is the eigenvector (up to scaling) associated with the zero eigenvalue of the singular matrix $C - \lambda_i M_C - (E - \lambda_i M_E)^T (B - \lambda_i M_B)^{-1} (E - \lambda_i M_E)$. By recalling the projection matrices $P_{\kappa_B} = V_{\kappa_B} V_{\kappa_B}^T M_B$ and $P_{\kappa_B}^{\perp} = I_d - V_{\kappa_B} V_{\kappa_B}^T M_B$, we can write $(B - \lambda_i M_B)^{-1}$ as

$$(B - \lambda_i M_B)^{-1} = P_{\kappa_B}(B - \lambda_i M_B)^{-1} + P_{\kappa_B}^{\perp}(B - \lambda_i M_B)^{-1}.$$

The eigenvector $x^{(i)}$ can be then expressed as the sum of the following three terms:

$$x^{(i)} = -\begin{bmatrix} P_{\kappa_B}(B-\lambda_i M_B)^{-1}(E-\lambda_i M_E)y^{(i)} \\ \end{bmatrix} - \begin{bmatrix} P^{\perp}_{\kappa_B}(B-\lambda_i M_B)^{-1}Ey^{(i)} \\ y^{(i)} \end{bmatrix} + \begin{bmatrix} \lambda_i P^{\perp}_{\kappa_B}(B-\lambda_i M_B)^{-1}M_E y^{(i)} \\ \end{bmatrix}. \tag{7}$$

Noticing that

$$P_{\kappa_B}(B-\lambda_i M_B)^{-1}(E-\lambda_i M_E)y^{(i)} \in \texttt{range}(V_{\kappa_B}),$$

we can write

$$x^{(i)} \in \texttt{range}\left(\begin{bmatrix} V_{\kappa_B} & P^{\perp}_{\kappa_B}(B-\lambda_i M_B)^{-1}Ey^{(i)} & \lambda_i P^{\perp}_{\kappa_B}(B-\lambda_i M_B)^{-1}M_E y^{(i)} \\ & y^{(i)} & \end{bmatrix}\right),$$

which, if we follow the notation in (5), leads to $x^{(i)} \in \mathcal{Z}_0 + \mathcal{Z}_i$, where

$$\mathcal{Z}_0 = \texttt{range}\left(\begin{bmatrix} V_{\kappa_B} \\ \end{bmatrix}\right), \quad \text{and}$$

$$\mathcal{Z}_i = \texttt{range}\left(\begin{bmatrix} P^{\perp}_{\kappa_B}(B-\lambda_i M_B)^{-1}Ey^{(i)} & \lambda_i P^{\perp}_{\kappa_B}(B-\lambda_i M_B)^{-1}M_E y^{(i)} \\ & y^{(i)} \end{bmatrix}\right).$$

Unfortunately, computing a basis of $\mathcal{Z}_i$ is impossible since it requires knowledge of both λ_i and $y^{(i)}$, $i = 1, \ldots, n_{ev}$. The rest of this section focuses on techniques to approximate $\mathcal{Z}_i$ without requiring the knowledge of neither λ_i nor $y^{(i)}$.

4.2 | Removing dependence on λ_i

Let $\kappa_1, \ldots, \kappa_\ell$, satisfy the inequality $\lambda_{n_{ev}} < \texttt{min}\left\{\delta^{(1)}_{\kappa_1+1}, \ldots, \delta^{(p)}_{\kappa_p+1}\right\}$. The matrix-valued function $P^{\perp}_{\kappa_B}(B-\zeta M_B)^{-1}$ is then analytic for any $\zeta \in \left(-\infty, \texttt{min}\left\{\delta^{(1)}_{\kappa_1+1}, \ldots, \delta^{(p)}_{\kappa_p+1}\right\}\right)$, and we can expand $P^{\perp}_{\kappa_B}(B-\lambda_i M_B)^{-1} = P^{\perp}_{\kappa_B}B^{-1}\sum_{j=0}^{\infty}\left[\lambda_i M_B B^{-1}\right]^j$, e.g., see[46,40]. The idea now is to remove the dependence on λ_i by truncating $P^{\perp}_{\kappa_B}(B-\lambda_i M_B)^{-1}$, albeit at the expense of an approximation. Similar approaches are discussed in[24,47] while a technique based on Chebyshev approximation is discussed in[48].

Let $\phi_\ell = \sum_{j=1}^{j=\ell-1} d_j$ and $\nu_\ell = \sum_{j=\ell+1}^{j=p} d_j$, denote the sum of interior variables of the substructures with an index lower and higher than ℓ, respectively. Then, for any $\psi \in \mathbb{N}$, the error introduced by approximating $P^{\perp}_{\kappa_B}(B-\lambda_i M_B)^{-1}$ by its ψ-th order expansion is equal to

$$P^{\perp}_{\kappa_B}\left((B-\lambda_i M_B)^{-1} - B^{-1}\sum_{j=0}^{j=\psi}\left[\lambda_i M_B B^{-1}\right]^j\right) = P^{\perp}_{\kappa_B}B^{-1}\sum_{j=\psi+1}^{\infty}\left[\lambda_i M_B B^{-1}\right]^j,$$

where the matrix on the right-hand side can be written as

$$P^{\perp}_{\kappa_B}B^{-1}\sum_{j=\psi+1}^{\infty}\left[\lambda_i M_B B^{-1}\right]^j = \sum_{\ell=1}^{\ell=p}\sum_{k=\kappa_\ell+1}^{k=d_\ell}\begin{bmatrix} 0_{\phi_\ell} \\ v_k^{(\ell)} \\ 0_{\nu_\ell}\end{bmatrix}\begin{bmatrix} 0_{\phi_\ell} & v_k^{(\ell)} & 0_{\nu_\ell}\end{bmatrix}^T\left[\sum_{j=\psi+1}^{\infty}\frac{\lambda_i^j}{\left(\delta_k^{(\ell)}-\lambda_i\right)\delta_k^{(\ell)j}}\right].$$

Proposition 1. Let $\lambda_{n_{ev}} < \texttt{min}\left\{\delta^{(1)}_{\kappa_1+1}, \ldots, \delta^{(p)}_{\kappa_p+1}\right\}$. Then, for any $i = 1, \ldots, n_{ev}$, and $\psi > 0$, the eigenvector $x^{(i)}$ can be written as

$$\begin{aligned} x^{(i)} = &-\begin{bmatrix} P_{\kappa_B}(B-\lambda_i M_B)^{-1}(E-\lambda_i M_E)y^{(i)} \\ \end{bmatrix} - \begin{bmatrix} P^{\perp}_{\kappa_B}B^{-1}\sum_{j=0}^{j=\psi}\left[\lambda_i M_B B^{-1}\right]^j Ey^{(i)} \\ y^{(i)} \end{bmatrix} \\ &+ \begin{bmatrix} \lambda_i P^{\perp}_{\kappa_B}B^{-1}\sum_{j=0}^{j=\psi-1}\left[\lambda_i M_B B^{-1}\right]^j M_E y^{(i)} \\ \end{bmatrix} \\ &+ O\left(\frac{\lambda_i^{\psi+1}}{\left(\texttt{min}\left\{\delta^{(1)}_{\kappa_1+1}, \ldots, \delta^{(p)}_{\kappa_p+1}\right\}-\lambda_i\right)\texttt{min}\left\{\delta^{(1)}_{\kappa_1+1}, \ldots, \delta^{(p)}_{\kappa_p+1}\right\}^{\psi}}\right\}, \end{aligned} \tag{8}$$

where the big-O term contains the asymptotic approximation error of $x^{(i)}$ as $\lambda_i \rightarrow 0$.

Proof. By expanding $P^{\perp}_{\kappa_B}(B-\lambda_i M_B)^{-1} = P^{\perp}_{\kappa_B} B^{-1} \sum\limits_{j=0}^{\infty} \left[\lambda_i M_B B^{-1}\right]^j$, we can write $x^{(i)}$ as

$$x^{(i)} = -\begin{bmatrix} P_{\kappa_B}(B-\lambda_i M_B)^{-1}(E-\lambda_i M_E)y^{(i)} \\ \\ \end{bmatrix} - \begin{bmatrix} P^{\perp}_{\kappa_B} B^{-1} \sum\limits_{j=0}^{\infty} \left[\lambda_i M_B B^{-1}\right]^j E y^{(i)} \\ y^{(i)} \end{bmatrix} + \begin{bmatrix} \lambda_i P^{\perp}_{\kappa_B} B^{-1} \sum\limits_{j=0}^{\infty} \left[\lambda_i M_B B^{-1}\right]^j M_E y^{(i)} \\ \\ \end{bmatrix}.$$

Truncating the resolvent series expansion of $(B-\lambda_i M_B)^{-1}$ introduces an error only along the top subvector of $x^{(i)}$. This error can be represented by a vector $z \in \mathbb{R}^d$,

$$\begin{aligned} z &= \lambda_i P^{\perp}_{\kappa_B} B^{-1} \sum_{j=\psi}^{\infty} \left[\lambda_i M_B B^{-1}\right]^j M_E y^{(i)} - P^{\perp}_{\kappa_B} B^{-1} \sum_{j=\psi+1}^{\infty} \left[\lambda_i M_B B^{-1}\right]^j E y^{(i)} \\ &= \lambda_i \sum_{\ell=1}^{\ell=p} \sum_{k=\kappa_\ell+1}^{k=d_\ell} \begin{bmatrix} 0_{\phi_\ell} \\ v_k^{(\ell)} \\ 0_{\nu_\ell} \end{bmatrix} \begin{bmatrix} 0_{\phi_\ell} & v_k^{(\ell)} & 0_{\nu_\ell} \end{bmatrix}^T \left[\sum_{j=\psi}^{\infty} \frac{\lambda_i^j}{\left(\delta_k^{(\ell)} - \lambda_i\right) \delta_k^{(\ell)j}} \right] M_E y^{(i)} \\ &- \sum_{\ell=1}^{\ell=p} \sum_{k=\kappa_\ell+1}^{k=d_\ell} \begin{bmatrix} 0_{\phi_\ell} \\ v_k^{(\ell)} \\ 0_{\nu_\ell} \end{bmatrix} \begin{bmatrix} 0_{\phi_\ell} & v_k^{(\ell)} & 0_{\nu_\ell} \end{bmatrix}^T \left[\sum_{j=\psi+1}^{\infty} \frac{\lambda_i^j}{\left(\delta_k^{(\ell)} - \lambda_i\right) \delta_k^{(\ell)j}} \right] E y^{(i)}. \end{aligned}$$

Since $\lambda_i < \mathtt{min}\left\{\delta^{(1)}_{\kappa_1+1}, \dots, \delta^{(p)}_{\kappa_p+1}\right\}$, the term $\dfrac{\lambda_i^j}{\left(\delta_k^{(\ell)} - \lambda_i\right) \delta_k^{(\ell)j}}$ is a decreasing function of k, $\forall \ell, j$. Moreover, as $\lambda_i \rightarrow 0$, the term $\dfrac{\lambda_i^{\psi+1}}{\left(\delta_k^{(\ell)} - \lambda_i\right) \delta_k^{(\ell)\psi}}$ is the slowest converging term of the resolvent series expansion. The result then follows directly. □

Equation (8) suggests that if either $\lambda_i \ll \mathtt{min}\left\{\delta^{(1)}_{\kappa_1+1}, \dots, \delta^{(p)}_{\kappa_p+1}\right\}$ or ψ is sufficiently large, it is reasonable to replace $P_{\kappa_B}(B-\lambda_i M_B)^{-1}$ by its ψ-term Taylor series truncation. However, since λ_i and $y^{(i)}$ are not available, we can not simply sum the three vector terms on the right-hand side of (8). Instead, we can approximate $x^{(i)}$ via a Rayleigh-Ritz projection onto the subspace spanned by the union of linear combinations that span the vectors in (8). This subspace is of the form

$$\hat{\mathcal{Z}}_i = \mathtt{range}\left(\begin{bmatrix} Z_0 & S_{i,E} & S_{i,M_E} \end{bmatrix}\right),$$

where the individual matrices are defined as

$$\begin{aligned} Z_0 &= \begin{bmatrix} V_{\kappa_B} \\ \\ \end{bmatrix}, \\ S_{i,E} &= \begin{bmatrix} P^{\perp}_{\kappa_B} B^{-1} E y^{(i)} & \cdots & P^{\perp}_{\kappa_B} B^{-1} (M_B B^{-1})^{\psi} E y^{(i)} \\ y^{(i)} & & \end{bmatrix}, \\ S_{i,M_E} &= \begin{bmatrix} P^{\perp}_{\kappa_B} B^{-1} M_E y^{(i)} & \cdots & P^{\perp}_{\kappa_B} B^{-1} (M_B B^{-1})^{\psi-1} M_E y^{(i)} \\ \\ \end{bmatrix}. \end{aligned} \tag{9}$$

The matrices $S_{i,E}$ and S_{i,M_E} have $\psi+1$ and ψ ($\psi \geq 1$) columns, respectively. Moreover, the subspace on the right-hand side of (9) is independent of λ_i.

Remark 1. Let $\psi \geq 1$ and $\mathtt{min}\left\{\delta^{(1)}_{\kappa_1+1}, \dots, \delta^{(p)}_{\kappa_p+1}\right\} = \gamma \lambda_i$ for some real $\gamma > 1$. Then, the error in the approximation of $x^{(i)}$ resulting by truncating $P^{\perp}_{\kappa_B}(B-\lambda_i M_B)^{-1}$ is of the order $O\left(\dfrac{1}{(\gamma-1)\gamma^{\psi}}\right)$.

4.3 | Removing dependence on $y^{(i)}$

The technique discussed in this section is a generalization of the approach described in[36] where it is suggested to build the Rayleigh-Ritz projection subspace by adding derivatives of eigenvectors associated with the n_{ev} smallest eigenvalues of the Schur complement matrix. In this section we extend this approach to eigenvector derivatives of the pencil $(S, -M_S)$.

Let us define the univariate $s \times s$ matrix-valued function $S : \zeta \in \mathbb{R} \rightarrow \mathbb{R}^{s \times s}$,

$$S(\zeta) = C - \zeta M_C - (E - \zeta M_E)^T (B - \zeta M_B)^{-1} (E - \zeta M_E).$$

The matrix-valued function $S(\zeta)$ is symmetric and analytic for any real $\zeta \notin \Lambda(B, M_B)$. Let now the univariate scalar-vector pair $(\theta_j, y_j) : \zeta \in \mathbb{R} \rightarrow (\theta_j(\zeta), y_j(\zeta)) \in \{\mathbb{R}, \mathbb{R}^s\}$ denote the jth eigenpair of the $s \times s$ symmetric generalized eigenvalue problem

$$S(\zeta) y(\zeta) = -\theta(\zeta) \frac{dS(\zeta)}{d\zeta} y(\zeta), \tag{10}$$

where the matrix-valued function

$$\begin{aligned} \frac{dS(\zeta)}{d\zeta} := dS(\zeta) = & - M_C - (E - \zeta M_E)^T (B - \zeta M_B)^{-1} M_B (B - \zeta M_B)^{-1} (E - \zeta M_E) \\ & + (E - \zeta M_E)^T (B - \zeta M_B)^{-1} M_E + M_E^T (B - \zeta M_B)^{-1} (E - \zeta M_E), \end{aligned}$$

denotes the derivative of the matrix-valued function $S(\zeta)$ with respect to ζ. Notice that $dS(0) = M_S$.

The matrix derivative $dS(\zeta)$ is symmetric and analytic for any real $\zeta \notin \Lambda(B, M_B)$. Moreover, $dS(\zeta)$ is negative definite for any $\zeta \notin \Lambda(B, M_B)$, and thus the eigenvalue problem in (10) is SPD. The eigenpair functions $(\theta_j(\zeta), y_j(\zeta))$, $j = 1, \ldots, s$, can be arranged so that they are analytic for any $\zeta \notin \Lambda(B, M_B)$[49,50]. In fact the latter holds for any $\zeta \in \mathbb{R}$ since the eigenvalues of the matrix pencil (B, M_B) are in fact eigenpoles of the eigenpairs $(\theta_j(\zeta), y_j(\zeta))$[37].

Following (6), the scalar $\lambda_i \notin \Lambda(B, M_B)$ is an eigenvalue of the pencil (A, M) if and only if the matrix pencil $(S(\lambda_i), -dS(\lambda_i))$ is singular. Thus, there exists an integer $j(i) \in \{1, 2, \ldots, s\}$ such that the eigenpair $(\theta_{j(i)}(\lambda_i), y_{j(i)}(\lambda_i))$ satisfies

$$\left[S(\lambda_i) + \theta_{j(i)}(\lambda_i) dS(\lambda_i) \right] y_{j(i)}(\lambda_i) = 0.$$

The eigenvector $y_{j(i)}(\lambda_i)$ associated with the eigenvalue $\theta_{j(i)}(\lambda_i)$ of zero modulus is equal (up to scaling) to the bottom $s \times 1$ vector $y^{(i)}$ of the eigenvector $x^{(i)}$ associated with the eigenvalue λ_i. The value of the subscript $j(i)$ is unknown, and can even repeat, e.g., when $s \leq n_{ev}$. Nonetheless, for typical problems of interest, the subscripts $j(i)$ are generally unique for each λ_i.

Expanding the analytic vector-valued function $y_{j(i)}(\zeta)$, $\zeta \in \left[0, \lambda_{n_{ev}}\right]$, through its Taylor series about the origin, and noticing $y_{j(i)}(\lambda_i) = y^{(i)}$ and $y_{j(i)}(0) = y_{j(i)}$, gives

$$y^{(i)} = y_{j(i)} + \sum_{k=1}^{\infty} \frac{\lambda_i^k}{k!} d^k y_{j(i)}, \tag{11}$$

where $d^k y_{j(i)} = \left(\frac{d^k y_{j(i)}(\zeta)}{d\zeta^k} \right)_{\zeta=0}$ denotes the kth derivative of the eigenvector $y_{j(i)}(\zeta)$, evaluated at the origin. The contribution of the kth derivative $d^k y_{j(i)}$ is weighted according to λ_i^k. Thus, assuming that each derivative $d^k y_{j(i)}(0)$, $k \in \mathbb{N}$, is bounded[2], the series in (11) generally converges faster for those eigenvalues of (A, M) which lie the closest to the origin, e.g., the n_{ev} smallest ones.

Proposition 2. Let $\lambda_{n_{ev}} < \texttt{min}\left\{\delta_{\kappa_1+1}^{(1)}, \ldots, \delta_{\kappa_p+1}^{(p)}\right\}$, and define the vector $\hat{y}_\tau^{(i)} = y_{j(i)} + \sum_{k=1}^{k=\tau} d^k y_{j(i)} \lambda_i^k / k!$. Then, for any $i = 1, \ldots, n_{ev}$, and $\psi > 0$, the eigenvector $x^{(i)}$ can be written as

$$\begin{aligned} x^{(i)} = & - \begin{bmatrix} P_{\kappa_B} (B - \lambda_i M_B)^{-1} (E - \lambda_i M_E) \hat{y}_\tau^{(i)} \\ \end{bmatrix} - \begin{bmatrix} P_{\kappa_B}^{\perp} B^{-1} \sum_{j=0}^{j=\psi} \left[\lambda_i M_B B^{-1}\right]^j E \hat{y}_\tau^{(i)} \\ \hat{y}_\tau^{(i)} \end{bmatrix} + \begin{bmatrix} \lambda_i P_{\kappa_B}^{\perp} B^{-1} \sum_{j=0}^{j=\psi-1} \left[\lambda_i M_B B^{-1}\right]^j M_E \hat{y}_\tau^{(i)} \\ \end{bmatrix} \\ & + O\left(\frac{\lambda_i^{\psi+1}}{\left(\texttt{min}\left\{\delta_{\kappa_1+1}^{(1)}, \ldots, \delta_{\kappa_p+1}^{(p)}\right\} - \lambda_i\right) \texttt{min}\left\{\delta_{\kappa_1+1}^{(1)}, \ldots, \delta_{\kappa_p+1}^{(p)}\right\}^{\psi}} \right) + O\left(\frac{\lambda_i^{\tau+1}}{\texttt{min}\left\{(\delta_{\kappa_1+1}^{(1)}, \ldots, \delta_{\kappa_p+1}^{(p)}\right\} - \lambda_i} \right), \end{aligned}$$

[2] By this we mean that each individual entry of $d^k y_{j(i)}(0)$ is bounded by a constant which is independent of k.

where the big-O term contains the asymptotic approximation error of $x^{(i)}$ as $\lambda_i \rightarrow 0$.

Proof. The first big-O term is obtained exactly as in Proposition 1. The proof follows by expanding $P_{\kappa_B}^{\perp}(B - \lambda_i M_B)^{-1} = P_{\kappa_B}^{\perp} B^{-1} \sum_{j=0}^{\infty} \left[\lambda_i M_B B^{-1}\right]^j$, and noticing that the term $P_{\kappa_B}(B - \lambda_i M_B)^{-1}(E - \lambda_i M_E)\hat{y}_{\tau}^{(i)}$ now introduces an error that is proportional to $\lambda_i^{\tau+1}$. □

The above expression tells us that approximating the exact eigenvector component $y^{(i)}$ by its truncation $\hat{y}_{\tau}^{(i)}$ results in an additional error of the order $O\left(\lambda_i^{\tau+1}\right)$. This error term will be of the same order with the one in (8) as long as $\psi = \tau$. Therefore, increasing ψ or τ disproportionately from the other, e.g., as in[24], does not lead to large asymptotic improvements.

In practice, the vector $\hat{y}_{\tau}^{(i)}$ can not be formed explicitly due to λ_i being unknown. However, since $\hat{y}_{\tau}^{(i)} \in \texttt{span}\left(y_{j(i)}, \ldots, d^{\tau} y_{j(i)}\right)$, we set $Y_{\{i,\tau\}} = \left[y_{j(i)}, \ldots, d^{\tau} y_{j(i)}\right]$, and update (9) as

$$
\begin{aligned}
Z_0 &= \begin{bmatrix} V_{\kappa_B} \\ \\ \end{bmatrix}, \\
S_{\{i,\tau,\psi\},E} &= \begin{bmatrix} P_{\kappa_B}^{\perp} B^{-1} M_B B^{-1} E Y_{\{i,\tau\}} & \cdots & P_{\kappa_B}^{\perp} B^{-1} (M_B B^{-1})^{\psi} E Y_{\{i,\tau\}} \\ & Y_{\{i,\tau\}} & \end{bmatrix}, \\
S_{\{i,\tau,\psi\},M_E} &= \begin{bmatrix} P_{\kappa_B}^{\perp} B^{-1} M_E Y_{\{i,\tau\}} & \cdots & P_{\kappa_B}^{\perp} B^{-1} (M_B B^{-1})^{\psi-1} M_E Y_{\{i,\tau\}} \\ & & \end{bmatrix}.
\end{aligned}
\tag{12}
$$

The matrices $S_{\{i,\tau,\psi\},E}$ and $S_{\{i,\tau,\psi\},M_E}$ have $(\tau + 1)(\psi + 1)$ and $(\tau + 1)\psi$ $(\psi \geq 1)$ columns, respectively. Moreover, they do not depend on neither λ_i nor $y^{(i)}$.

4.4 | The proposed algorithm

Algorithm 4.1 summarizes the proposed eigenvalue solver for computing the n_{ev} smallest eigenvalues and associated eigenvectors of the matrix pencil (A, M) such that $\lambda_{n_{ev}} < \beta < \lambda_{n_{ev}+1}$ where $\beta \in \mathbb{R}$ is a known scalar. The Rayleigh-Ritz projection matrix is equal to

$$
Z = \left[Z_0 \; Z_1 \; \cdots \; Z_{n_{ev}}\right], \tag{13}
$$

where

$$
Z_0 = \begin{bmatrix} V_{\kappa_B} \\ \\ \end{bmatrix}, \quad Z_i = \left[S_{\{i,\tau,\psi\},E} \quad S_{\{i,\tau,\psi\},M_E}\right].
$$

By default, we set $\tau = \psi = 1$, since according to the analysis presented in Propositions 1 and 2 these values are sufficient to lead to a quadratic error in the approximation of each sought eigenvector.

The parameters $\kappa_1, \ldots, \kappa_p$, are typically set on-the-fly so that $\delta_{\kappa_{\ell}+1}^{(\ell)}$ corresponds to a scalar multiple of an upper bound of the largest sought eigenvalue $\lambda_{n_{ev}}$. Nonetheless, determining an optimal choice for these values is not crucial, as we expect most of the accuracy in the approximation of the n_{ev} sought eigenpairs of (A, M) to stem by the resolvent approximation in (9). Therefore, we can stop computing additional eigenpairs of the pencil $(B_{\ell}, M_{B_{\ell}})$, $\ell = 1, \ldots, p$, once $\delta_j^{(\ell)}$ surpasses the upper bound of $\lambda_{n_{ev}}$, since the matrix $(B - \zeta M_B)^{-1}$ is then analytic in the interval $\zeta \in [-\infty, \ldots, \lambda_{n_{ev}}]$. Thus, the values of $\kappa_1, \ldots, \kappa_p$, can be regarded as optional. The same idea applies to the pencil $(S, -M_S)$ for which we set $\kappa_S = n_{ev}$.

The total memory cost of Algorithm 4.1 for the default options is equal to $n_{ev}(2n + 4d) + \sum_{\ell=1}^{\ell=p} \kappa_{\ell} d_{\ell}$, leading to a Rayleigh-Ritz eigenvalue problem of size $6n_{ev} + \sum_{\ell=1}^{\ell=p} \kappa_{\ell}$. When M is diagonal, we have $M_E = 0$ and the total memory cost reduces to $n_{ev}(2n + 2d) + \sum_{\ell=1}^{\ell=p} \kappa_{\ell} d_{\ell}$. The size of the Rayleigh-Ritz eigenvalue problem is then equal to $4n_{ev} + \sum_{\ell=1}^{\ell=p} \kappa_{\ell}$.

The computational complexity to solve the Rayleigh-Ritz eigenvalue problem is cubic with respect to the dimension projection subspace. Therefore, if we choose $\sum_{\ell=1}^{\ell=p} \kappa_{\ell} = O(n_{ev})$, the cost of this step runs at $O(n_{ev}^3)$.

Remark 2. The Rayleigh-Ritz projection matrix produced by Algorithm 4.1 with parameter values $\psi = 1$, $\tau = 1$, and $\kappa_S = n_{ev}$, can be expressed as

$$
Z = \begin{bmatrix} V_{\kappa_B} & P_{\kappa_B}^{\perp} B^{-1} E Y & P_{\kappa_B}^{\perp} B^{-1} M_E Y & P_{\kappa_B}^{\perp} B^{-1} M_B B^{-1} E Y \\ & & Y & \end{bmatrix},
$$

where $Y = [y_1, \ldots, y_{n_{ev}}, dy_1, \ldots, dy_{n_{ev}}]$. When $M_E = 0$, we have $P_{\kappa_B}^{\perp} B^{-1} M_E Y = 0$.

ALGORITHM **4.1.** *Enhanced substructuring algorithm*

```
0a. Input: $A$, $M$, $n_{ev} \in \mathbb{N}$, (optionally) $\kappa_1, \dots, \kappa_p, \kappa_S \in \mathbb{N}$
0b. Reorder matrices $A$ and $M$ as in (1)
1a. For $\ell = 1, \dots, p$
2.      Compute $V_{\kappa_\ell,\ell} = \texttt{eigs}\left(B_\ell, M_{B_\ell}, \kappa_\ell\right)$   ($\kappa_\ell$ is determined on-the-fly)
1b. End
3.  Set $V_{\kappa_B} = \texttt{blkdiag}\left(V_{\kappa_1,1}, \dots, V_{\kappa_p,p}\right)$
4.  If $\kappa_S$ undefined, set $\kappa_S = n_{ev}$ and compute $V_{\kappa_S} = \texttt{eigs}\left(S, -M_S, n_{ev}\right)$
5.  Set $\tau = 1$ and compute $dy_1, \dots, dy_{n_{ev}}$ by (A4)
6.  Form matrix $Z$ in (13) using $\psi = 1$
7.  Compute the eigenpairs $\left(\widetilde{\lambda}_i, \widetilde{f}^{(i)}\right)$ associated with the $n_{ev}$ smallest
    eigenvalues of the Rayleigh-Ritz pencil $(Z^T AZ, Z^T MZ)$
8.  Approximate the eigenpair $\left(\lambda_i, x^{(i)}\right)$, $1 \leq i \leq n_{ev}$, of the pencil $(A, M)$
    by $\left(\widetilde{\lambda}_i, Z\widetilde{f}^{(i)}\right)$
```

4.5 | Multilevel extensions

For high-dimensional discretizations, e.g., three-dimensional or higher, single-level partitionings with a large number of substructures can lead to very large Schur complement matrices. An alternative is to reduce the number of substructures and consider multilevel partitionings. More specifically, assume a multilevel setting with $l_{ev} \in \mathbb{N}$ levels. Each algebraic substructure associated with interior valriables at level $1 \leq k \leq l_{ev}$ is then further partitioned into: *a*) p algebraic substructures, and *b*) their edge separator. Figure 3 depicts a recursive partitioning of the 4×10 grid shown in Figure 1 . Here, we set $l_{ev} = 2$ and $p = 2$. Dashed lines denote the edge-separator $\mathcal{I}_0$ associated with the first level. The algebraic substructures Ω_1 and Ω_2 are further partitioned into two algebraic substructures each, denoted by $\Omega_{i,1}$ and $\Omega_{i,2}$, and their respective edge separator $\mathcal{I}_1$ and $\mathcal{I}_2$, denoted by the dotted lines.

Algorithm 4.1 can be combined with multilevel partitionings by performing Step 2 in Algorithm 4.1 through applying itself recursively instead of shift-and-invert IRL. This recursion continues until the maximum given level l_{ev} is reached. However, instead of returning approximate eigenpairs, the application of Algorithm 4.1 at the non-root levels returns the actual Rayleigh-Ritz projection matrix.

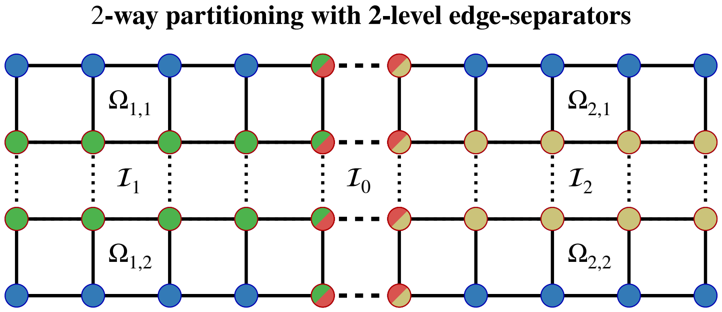

FIGURE 3 Partitioning of a 4×10 grid into four subgraphs using nested edge-separators.

In terms of matrix sparsity pattern, let the graph associated with the matrix $|B_\ell| + |M_{B_\ell}|$, $1 \leq \ell \leq p$, be further partitioned into p algebraic substructures. The rows and columns of the $d_\ell \times d_\ell$ matrices B_ℓ (similarly for M_{B_ℓ}) can be then permuted such

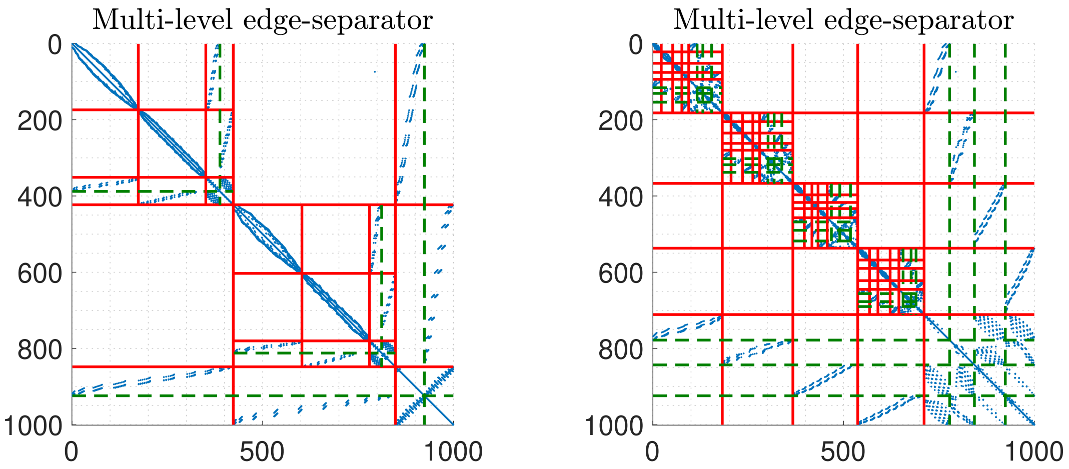

FIGURE 4 Sparsity pattern of the problem shown in Figure 2 using a two-level partitioning. Left: $p = 2$. Right: $p = 4$.

that

$$B_\ell = \begin{array}{c} \\ d_1^{(\ell)} \\ \vdots \\ d_p^{(\ell)} \\ s^{(\ell)} \end{array} \begin{array}{c} \begin{array}{cccc} d_1^{(\ell)} & \cdots & d_p^{(\ell)} & s^{(\ell)} \end{array} \\ \begin{pmatrix} B_1^{(\ell)} & & & E_1^{(\ell)} \\ & \ddots & & \vdots \\ & & B_p^{(\ell)} & E_p^{(\ell)} \\ \left(E_1^{(\ell)}\right)^T & \cdots & \left(E_p^{(\ell)}\right)^T & C_\ell \end{pmatrix} \end{array},$$

where $d_\ell = d_1^{(\ell)} + \cdots + d_p^{(\ell)} + s^{(\ell)}$.

Figure 4 plots the sparsity pattern of the matrix considered in Figure 2 , where the graph of $\mathcal{F}(A) + \mathcal{F}(M)$ is now partitioned into $l_{ev} = 2$ levels. Notice that interface variables at at the root level are not affected by partitionings at the second level since we only partition interior variables.

4.5.1 | Rayleigh-Ritz projection matrix for two-level partitionings

As an example, consider a two-level partitioning with p algebraic substructures at each level. In this case, the Rayleigh-Ritz projection matrix Z associated with the pencil (A, M) becomes

$$Z = \begin{bmatrix} Z_r & B^{-1}EY & B^{-1}M_E Y & B^{-1}M_B B^{-1}EY \\ & Y & & \end{bmatrix}, \quad Z_r = \mathtt{blkdiag}\left(Z_1^{(1)}, \ldots, Z_p^{(1)}\right),$$

where $Z_\ell^{(1)}$ denotes the Rayleigh-Ritz projection matrix returned by applying Algorithm 4.1 to approximate the κ_ℓ sought eigenvectors of the pencil (B_ℓ, M_{B_ℓ}), $\ell = 1, \ldots, p$.

The computational cost of the multilevel variant of Algorithm 4.1 depends on the value of κ_ℓ, $\ell = 1, \ldots, p$. In particular, if $\mathcal{T}(n_{ev})$ denotes the total computational cost of Algorithm 4.1 using the default settings, the total computational cost of the multilevel variant is bounded by $(p+1)\mathcal{T}(\max_{\ell=1,\ldots,p} \kappa_\ell)$. As a result, the value of κ_ℓ is generally chosen inversely proportional to the number of levels/subdomains. In the next section we demonstrate that even when $\max_{\ell=1,\ldots,p} \kappa_\ell \ll n_{ev}$, Algorithm 4.1 can approximate the eigenpairs of the pencil (A, M) up to several digits of accuracy.

5 | NUMERICAL EXPERIMENTS

Our experiments are conducted in a Matlab environment (version R2020b), using 64-bit arithmetic, on a single core of a computing system equipped with an 2.3 GHz 8-Core Intel Core i9 processor and 64 GB of DDR4 system memory. Throughout this section we are interested in computing the eigenpairs associated with the n_{ev} smallest eigenvalues of the pencil (A, M). Unless mentioned otherwise, the number of sought eigenpairs will be set equal to $n_{ev} = 100$.

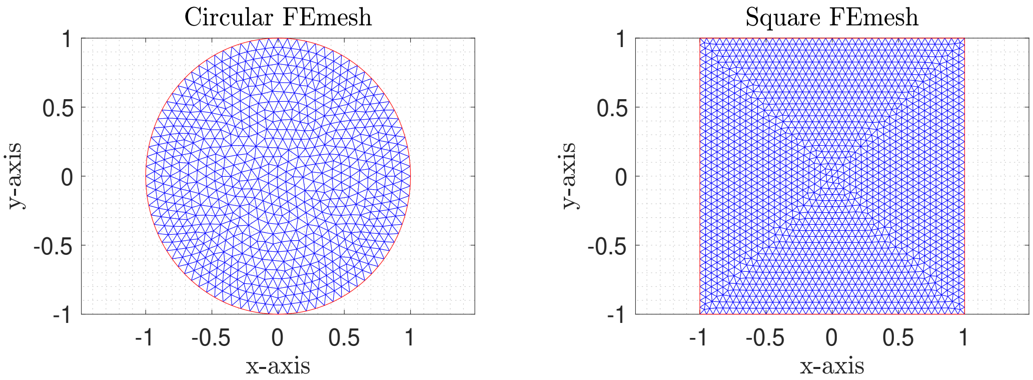

FIGURE 5 Left: FE triangulation of a circular domain. Right: FE triangulation of a square domain.

The residual norm of an approximate eigenpair $(\widetilde{\lambda}_i, \widetilde{x}^{(i)})$ of the true eigenpair $(\lambda_i, x^{(i)})$ is equal to $\left\|A\widetilde{x}^{(i)} - \widetilde{\lambda}_i M\widetilde{x}^{(i)}\right\|_2 / \|\widetilde{x}^{(i)}\|_M$, while the relative eigenvalue error is computed as $\left|\widetilde{\lambda}_i - \lambda_i\right| / |\lambda_i|$.

The matrices A and M considered throughout this section are derived from discretizations of the Dirichlet eigenvalue problem $\Delta u + \lambda u = 0$ in some domain Ω, where Δ denotes the Laplace operator and $u_{|\partial\Omega} = 0$. The domain Ω is set to: a) a circular domain centered at the origin with a radius of one, and b) a square domain centered at the origin with edge size of one. For the square domain, we consider both linear Finite Element (FE) and Finite Difference (FD) discretizations. A snapshot of the triangulation of each domain, using a target maximum edge size of 0.5 (circular mesh) and 0.1 (square mesh), is shown in Figure 5 . For the FD discretization, we assume a 506×296 grid of the unit square. Further details can be found in Table 1 , while the size of the Schur complement matrix as p varies is listed in Table 2 . While not actively pursued throughout our experiments, Algorithm 4.1 can be also applied to pencils stemming from adaptive finite element approaches[51,52,53].

TABLE 1 n: size of (A, M), $nnz(.)$: number of nonzero entries. The expression $s(p)$ returns the number of interface variables for the corresponding value of p.

#	Matrix pencil	n	$nnz(A)/n$	$nnz(M)/n$
1.	Square FE mesh	45,064	6.96	6.96
2.	Circular FE mesh	37,381	11.4	11.4
3.	Rectangular FD mesh	149,766	4.99	1.00

TABLE 2 Size of the Schur complement matrix as a function of p.

Matrix pencil	n	$p = 4$	$p = 8$	$p = 16$	$p = 32$	$p = 64$
Square FE mesh	45,064	864	1,701	2,662	4,098	6,148
Circular FE mesh	37,381	854	1,554	2,522	3,826	5,666
Rectangular FD mesh	149,766	1,878	2,990	5,036	7,466	11,288

5.1 | Computational aspects of Algorithm 4.1

5.1.1 | Eigenvalues of the pencils (B, M_B) and $(S, -M_S)$ as p varies

As the number of algebraic substructures increases, the number of interior (interface) variables decreases (increases) and vice versa. A natural question is how the latter affects the spectrum of the pencils $\left(S, -M_S\right)$ and (B_ℓ, M_{B_ℓ}), $\ell = 1, \ldots, p$. While this

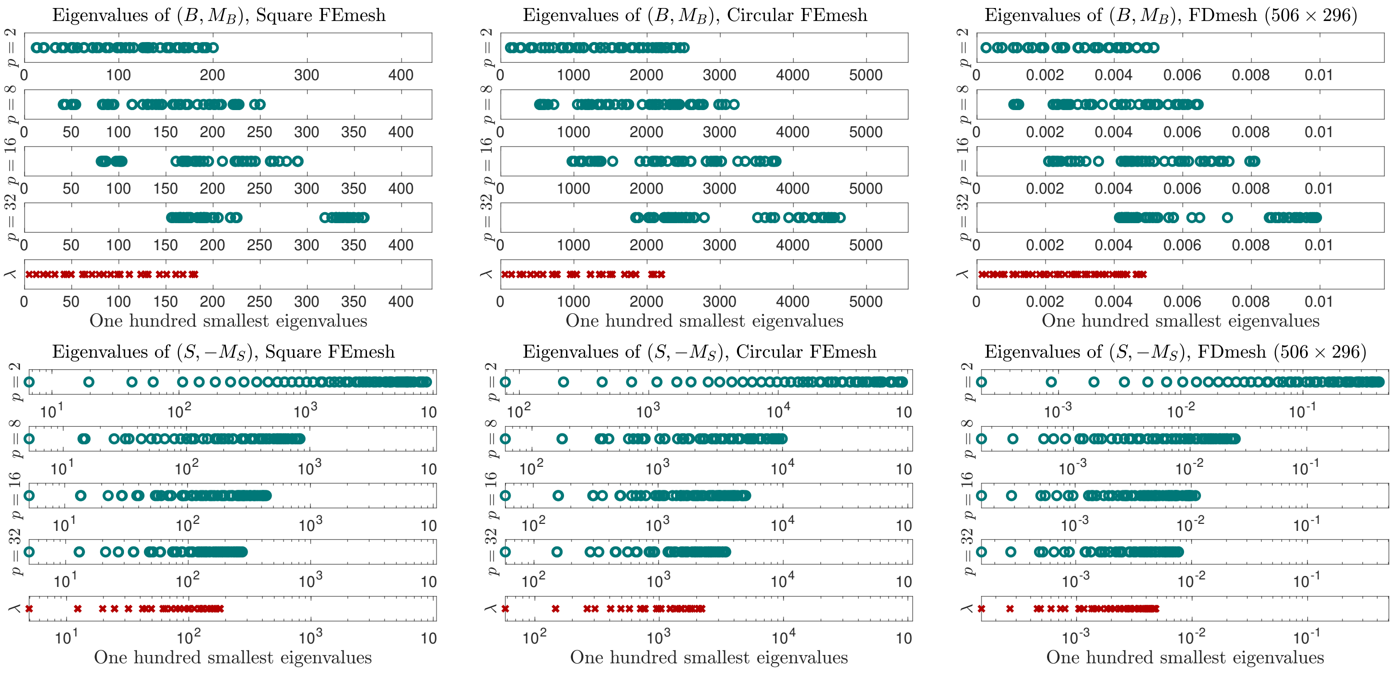

FIGURE 6 Plot of the $n_{ev} = 100$ smallest eigenvalues of the pencils (B, M_B) and $(S, -M_S)$ as p varies. The eigenvalues $\lambda_1, \dots, \lambda_{n_{ev}}$, are also plotted as a reference.

can not be answered in general, a direct extensions of Cauchy's interlacing property to symmetric (Hermitian) matrix pencils (see[54]) shows that the eigenvalues of the pencils $\left(S, -M_S\right)$ and (B, M_B) satisfy the inequalities

$$\lambda_i \le \theta_i \le \lambda_{n-s+i}, \quad \text{and} \quad \lambda_j \le \delta_j \le \lambda_{n-d+j}, \tag{14}$$

where $i = 1, \dots, s$, and $j = 1, \dots, d$. Therefore, increasing the number of algebraic substructures is more likely to lead to pencils (B_ℓ, M_{B_ℓ}), $\ell = 1, \dots, p$, whose smallest eigenvalues might lie further away from the origin, in which case κ_ℓ can be chosen smaller. Similarly, decreasing the number of algebraic substructures is more likely to lead to a pencil $(S, -M_S)$ whose smallest eigenvalues lie further away from the origin, and thus κ_S can be chosen smaller. Thus, setting proper values of $\kappa_1, \dots, \kappa_p$, and κ_S, depends non-trivially on the graph partitioning of the graph associated with $\mathcal{F}(A) + \mathcal{F}(M)$.

Figure 6 shows the $n_{ev} = 100$ algebraically smallest eigenvalues of the pencils (B, M_B) and $(S, -M_S)$ for $p \in \{2, 8, 16, 32\}$. Smaller values of p lead to fewer interface variables and a much higher spread of the smallest eigenvalues of the pencil $(S, -M_S)$ compared to when p increases. For example, $\theta_{n_{ev}}$ is about fifty times larger than $\lambda_{n_{ev}}$ when $p = 2$ but less than twice as large as $\lambda_{n_{ev}}$ when $p = 32$. Therefore, κ_S might be chosen smaller when p is small. The same observations can be made about the smallest eigenvalues of the pencil (B, M_B). As we observe experimentally, $p = 2$ is the optimal choice regarding the pencil $(S, -M_S)$ as its eigenvalues are relatively further away from the origin. On the other hand, $p = 2$ is the most challenging case with respect to the pencil (B, M_B), in the sense that its smallest eigenvalues are relatively closer to the origin compared to larger values of p.

5.1.2 | Cost of computing eigenvectors of the pencil $(S, -M_S)$

We now focus on the computational cost associated with applying shift-and-invert IRL to compute the κ_S eigenpairs of the pencil $(S, -M_S)$ stemming from partitioning a regular grid discretization into p perfectly balanced algebraic substructures using edge separators. Shift-and-invert requires the factorization of the Schur complement matrix S where the size of the latter is bounded by $s = p\sqrt{n}$ (2D) and $s = pn^{2/3}$ (3D). The Cholesky factorization cost then runs at $O(p^3 n^{3/2})$ and $O(p^3 n^2)$, respectively, while orthogonalization cost runs at $O(p\sqrt{n}\kappa_S^2)$ and $O(pn^{2/3}\kappa_S^2)$, respectively. For reference, the orthogonalization cost of applying shift-and-invert IRL directly to the pencil (A, M) runs at $O(nn_{ev}^2)$.

Table 3 lists the number of iterations required by shift-and-invert IRL to compute the eigenpairs associated with the κ_S smallest eigenvalues of the pencils (A, M) and $(S, -M_S)$, as p varies, for the problem "FDmesh". We also report the amount

TABLE 3 Number of iterations ("iter") performed by shift-and-invert IRL to compute the eigenpairs associated with the κ_S smallest eigenpairs of the pencils (A, M) and $(S, -M_S)$ for different values of p (values reported stand for the problem "506×296 FDmesh"). The amount of time spent on solving linear systems with the Schur complement matrix ("sol"), and orthogonalizing the Krylov subspace basis ("orth"), are reported separately.

	$\kappa_S = 100$			$\kappa_S = 200$			$\kappa_S = 300$		
	iter	sol	orth	iter	sol	orth	iter	sol	orth
$p = 4$ $(s = 939)$	201	0.40	0.01	401	0.97	0.06	601	1.53	0.13
$p = 16$ $(s = 2,518)$	201	1.24	0.04	401	2.65	0.15	601	4.15	0.35
$p = 64$ $(s = 5,644)$	251	2.53	0.13	401	4.70	0.34	601	7.76	0.95
(A, M)	301	5.92	3.94	501	10.8	11.5	751	16.7	27.6

of time spent on (as returned by Matlab's internal timer): *a*) solving linear systems with the Schur complement matrix S, and *b*) maintaining orthogonality of the Krylov subspace. As we can see from the table, substructuring can lead to a considerable reduction in orthogonalization costs compared to applying shift-and-invert IRL to (A, M). This reduction is owed to the smaller number of IRL iterations performed, as well as the application of orthogonalization to much shorter vectors than the size of the pencil (A, M). The latter becomes more pronounced when the size of the pencil $(S, -M_S)$ grows slowly with increasing values of p, e.g., discretizations of 2D domains. Similar results are also observed for the FE pencils listed in Table 1 .

5.2 | Benchmarking Algorithm 4.1

Figure 7 plots the relative eigenvalue errors and corresponding residual norms achieved by Algorithm 4.1 when: *a*) $\kappa_S \in \{n_{ev}, 3n_{ev}, 5n_{ev}\}$ and $\tau = 0$, and *b*) $\kappa_S = n_{ev}$ and $\tau = 1$. The first option is equivalent to a p-way generalization of AMLS, while the second option is equivalent to Algorithm 4.1. For both approaches we vary the number of partitions and computed interior eigenmodes as $(\kappa, p) \in \{(4, 64), (16, 16)\}$, where $\kappa = \kappa_1 = \ldots = \kappa_p$. In summary, exploiting eigenvector derivatives can lead to similar or higher accuracy than that achieved by choosing as high as $\kappa_S = 5n_{ev}$. Nonetheless, the main advantage of Algorithm 4.1 over previous substructuring algorithms, e.g., AMLS, is not as much its higher accuracy as that it achieves the latter while computing only n_{ev} eigenpairs of the pencil $(S, -M_S)$.

Figure 8 plots the relative eigenvalue errors and corresponding residual norms achieved by Algorithm 4.1 for the same parameter choice as in the previous experiment, except that now we approximate $P^{\perp}_{\kappa_B}(B - \lambda_i M_B)^{-1}$ using both $\psi = 0$ and $\psi = 1$. The results plotted stand for the pencil "Square FEmesh" but similar behavior was observed for the rest of our test problems. In summary, increasing the value of κ_S (or τ) when $\psi = 0$ does not generally lead to major accuracy improvements due to the inaccuracies in the part of the projection subspace associated with interior variables.

5.2.1 | Comparisons against second-order linearizations

Recall that $M_S = dS(0)$, and thus the eigenvalue problem associated with the pencil $(S, -M_S)$ is equivalent to a first-order linearization of the nonlinear eigenvalue problem $S(\zeta)y = 0$ for $\zeta = 0$. An idea suggested in[24] is to replace $\texttt{span}\left(y_1, \ldots, y_{\kappa_S}\right)$ by $\texttt{span}\left(w_1, \ldots, w_{\kappa_S}\right)$, where $w_j \in \mathbb{R}^s$ denotes the eigenvector associated with the jth smallest positive eigenvalue of the generalized eigenvalue problem $\begin{bmatrix} 0 & I_s \\ S & -M_S \end{bmatrix} \begin{bmatrix} w \\ \gamma w \end{bmatrix} = \gamma \begin{bmatrix} I_s & 0 \\ 0 & d^2S(0) \end{bmatrix} \begin{bmatrix} w \\ \gamma w \end{bmatrix}$. This pencil is a second-order linearization of the nonlinear eigenvalue problem $S(\zeta)y = 0$, and leads to greater accuracy compared to using eigenvectors of the first-order linearization $(S, -M_S)$. On the other hand, the size of the former eigenvalue problem is twice as large.

To compare the approach suggested in this paper against second-order linearizations, we consider a scenario where the eigenvector portions associated with interior variables are known explicitly, i.e., we pre-compute the top $d \times 1$ subvectors of each eigenvector $x^{(i)} = \begin{bmatrix} u^{(i)} \\ y^{(i)} \end{bmatrix}$, $i = 1, \ldots, n_{ev}$, and inject them into the Rayleigh-Ritz projection subspace. We then perform a Rayleigh-Ritz projection where we set the projection subspace associated with interface variables equal to $\texttt{span}\left(w_1, w_2, \ldots, w_{\kappa_S}\right)$ (for the second-order linearization) and $\texttt{span}\left(y_1, y_2, \ldots, y_{n_{ev}}, dy_1, dy_2, \ldots, y_{n_{ev}}\right)$ (for the proposed algorithm), respectively.

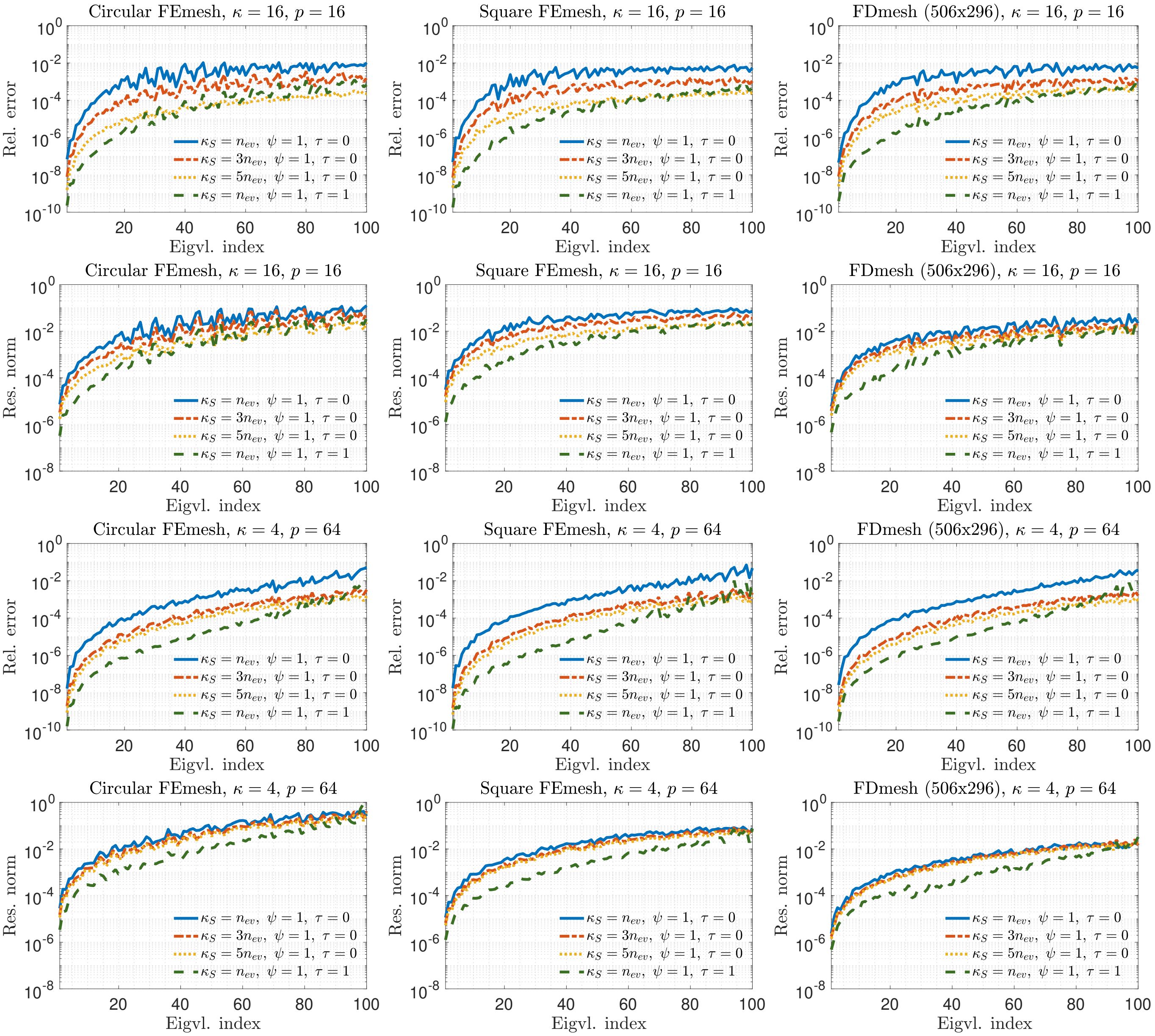

FIGURE 7 Plot of relative eigenvalue errors and corresponding residual norms achieved by Algorithm 4.1 for different combinations of κ and p such that κp is constant.

Figure 9 presents the relative eigenvalue errors and corresponding residual norms achieved by the two different projection matrices when $p = 8$. The number of computed eigenvectors in the second-order linearization is varied as $\kappa_S \in \{n_{ev}, 3n_{ev}, 5n_{ev}\}$. In summary, exploiting eigenvector derivatives leads to an accuracy which is similar to that achieved by setting $\kappa_S = 3n_{ev}$ but lower than that achieved by setting $\kappa_S = 5n_{ev}$, especially for those eigenpairs associated with eigenvalues located further away from the origin. From a computational perspective, the actual number of computed eigenpairs of the second-order linearization is twice as large as κ_S due to the maximal indefiniteness of the pencil $\left(\begin{bmatrix} 0 & I_s \\ S & -M_S \end{bmatrix}, \begin{bmatrix} I_s & 0 \\ 0 & d^2S(0) \end{bmatrix}\right)$. Moreover, as noted above, the size of this pencil is $2s \times 2s$. As a result, the computational cost of Algorithm 4.1 is generally much smaller than that of the second-order linearization.

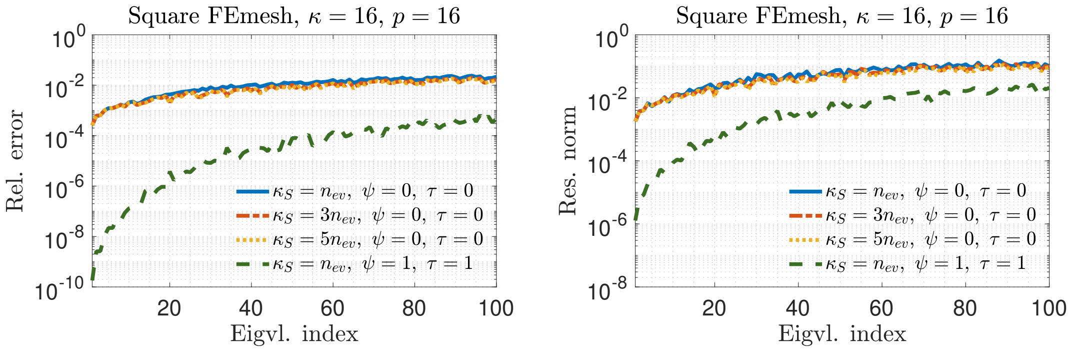

FIGURE 8 Plot of relative eigenvalue errors and corresponding residual norms achieved by Algorithm 4.1

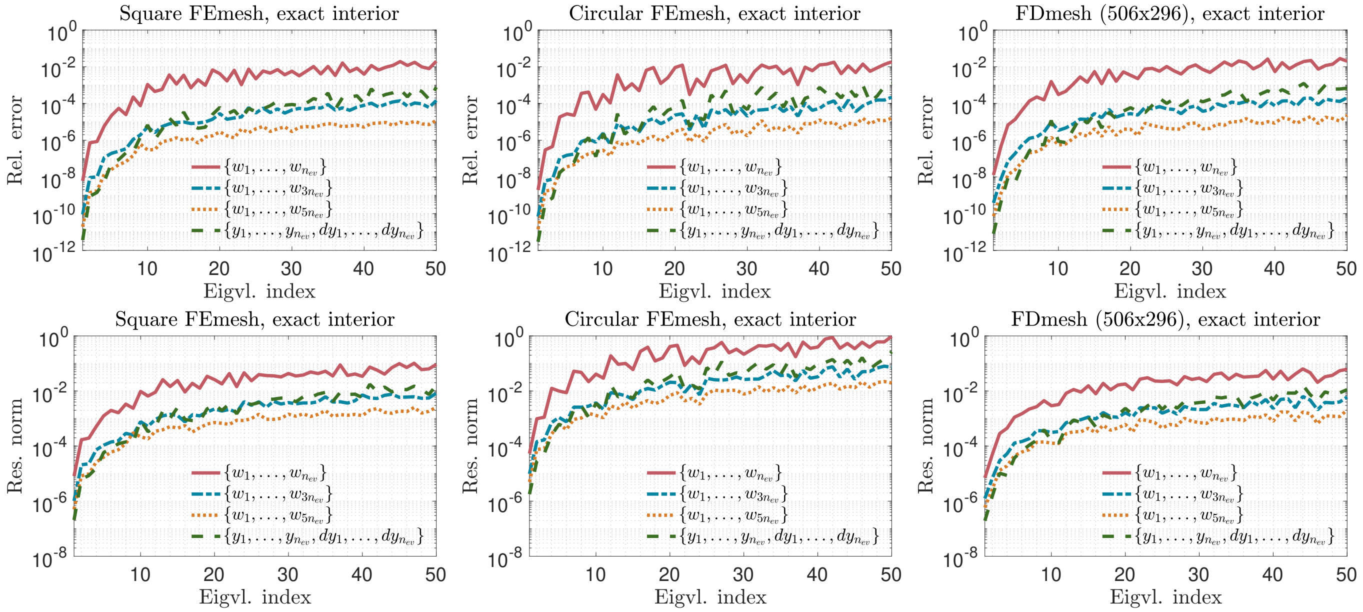

FIGURE 9 Comparing the interface projection subspace in Algorithm 4.1 against that of a second-order linearization of the spectral Schur complement.

5.3 | Two-level partitioning

Finally, we consider the accuracy of the approximate eigenpairs returned by the two-level variant of Algorithm 4.1 as the number of algebraic substructures at each level varies. Figure 10 plots the relative eigenvalue errors and corresponding residual norms returned by a two-level variant of Algorithm 4.1 as $p = \{2, 8, 16\}$ and the main parameters are chosen as: *a*) $\psi = 0$, $\tau = 0$, and $\kappa_S = 2n_{ev}$, and *b*) $\psi = 1$, $\tau = 1$, and $\kappa_S = n_{ev}$. For this set of experiments, the number of sought eigenpairs was set equal to $n_{ev} = 50$. The number of eigenpairs approximated by the nested application of Algorithm 4.1 to each matrix pencil (B_ℓ, M_{B_ℓ}), $\ell = 1, \ldots, p$, was set to $\kappa = 64$, 32, and $\kappa = 16$, for the cases $p = 2$, 8, and $p = 16$, respectively. In summary, using fewer levels leads to higher accuracy, especially for the case $\psi = \tau = 1$ since the subspace returned by the nested application of Algorithm 4.1 is richer in relevant spectral information.

6 | CONCLUSION

This paper presented an algebraic substructuring eigenvalue solver to approximate the algebraically smallest eigenvalues and corresponding eigenvectors of symmetric matrix pencils. The proposed technique divides the graph associated with the matrix

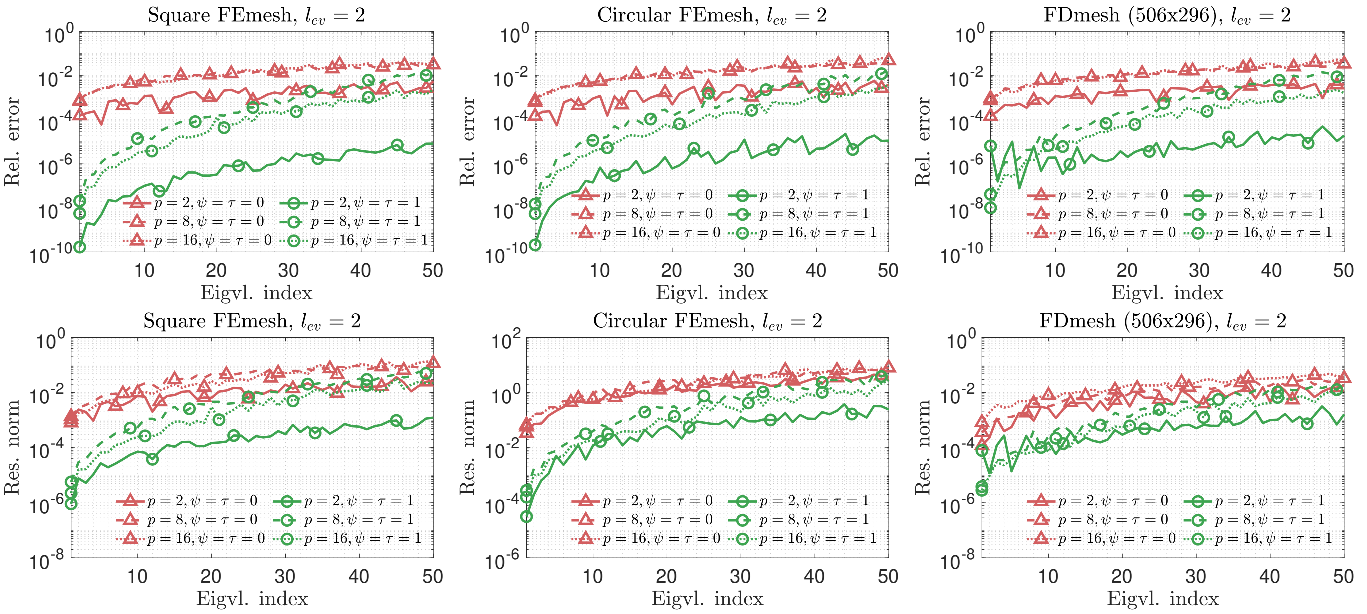

FIGURE 10 Relative eigenvalue errors and residual norms returned by a two-level variant of Algorithm 4.1.

$\mathcal{F}(A) + \mathcal{F}(M)$ into a number of algebraic substructures and builds a Rayleigh-Ritz projection subspace by combining spectral information associated with interior and interface variables. The subspace associated with interior variables is built by computing substructural eigenvectors and truncated Neumann series expansions of resolvent matrices. The subspace associated with interface variables is built by combining eigenmodes of linearized spectral Schur complements together with their leading derivatives. Several practical details were discussed, including extensions to multilevel settings, and analysis of the total computational cost. Experiments performed on problems stemming from discretizing the Dirichlet eigenvalue problem suggest that the proposed technique can achieve several digits of accuracy while considerably reducing orthogonalization costs encountered in standard applications of Krylov subspace eigenvalue solvers.

As part of future work, it would be interesting to develop a distributed memory version of the proposed algorithm in order to fully take advantage of the underlying p-way partitioning. Such an implementation will also help evaluating the proposed algorithm on real-world engineering problems. Another interesting direction is to reduce the computational complexity of the proposed algorithm through combinations with hierarchical matrix approximation techniques[55,56,57].

ACKNOWLEDGMENTS

Part of this work was completed under the support of the Herman H. Goldstine Postdoctoral Fellowship program of International Business Machines Corporation.

This study does not have any conflicts to disclose.

References

1. Arbenz P, Hetmaniuk UL, Lehoucq RB, Tuminaro RS. A comparison of eigensolvers for large-scale 3D modal analysis using AMG-preconditioned iterative methods. *International Journal for Numerical Methods in Engineering* 2005; 64(2): 204–236.

2. Bathe KJ. *Solution methods for large generalized eigenvalue problems in structural engineering*. National Technical Information Service, US Department of Commerce . 1971.

3. Bathe KJ. The subspace iteration method–revisited. *Computers & Structures* 2013; 126: 177–183.

4. Bathe KJ, Ramaswamy S. An accelerated subspace iteration method. *Computer Methods in Applied Mechanics and Engineering* 1980; 23(3): 313–331.

5. Kim JG, Boo SH, Lee PS. An enhanced AMLS method and its performance. *Computer Methods in Applied Mechanics and Engineering* 2015; 287: 90–111.

6. Ko JH, Bai Z. High-frequency response analysis via algebraic substructuring. *International Journal for Numerical Methods in Engineering* 2008; 76(3): 295–313.

7. Komzsik L, Rose T. Substructuring in MSC/NASTRAN for large scale parallel applications. *Computing Systems in Engineering* 1991; 2(2-3): 167–173.

8. Ng AY, Jordan MI, Weiss Y. On spectral clustering: Analysis and an algorithm. In: ; 2002: 849–856.

9. Kronik L, Makmal A, Tiago ML, et al. PARSEC–the pseudopotential algorithm for real-space electronic structure calculations: recent advances and novel applications to nano-structures. *physica status solidi (b)* 2006; 243(5): 1063–1079.

10. Parlett BN. *The symmetric eigenvalue problem*. SIAM . 1998.

11. Bai Z, Demmel J, Dongarra J, Ruhe A, Vorst v. dH. *Templates for the solution of algebraic eigenvalue problems: a practical guide*. SIAM . 2000.

12. Saad Y. *Numerical methods for large eigenvalue problems: revised edition*. 66. SIAM . 2011.

13. Lehoucq RB, Sorensen DC, Yang C. *ARPACK Users' Guide: solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods*. SIAM . 1998.

14. Saad Y. *Iterative methods for sparse linear systems*. SIAM . 2003.

15. Wu K, Simon H. Thick-restart Lanczos method for large symmetric eigenvalue problems. *SIAM Journal on Matrix Analysis and Applications* 2000; 22(2): 602–616.

16. Grimes RG, Lewis JG, Simon HD. A shifted block Lanczos algorithm for solving sparse symmetric generalized eigenproblems. *SIAM Journal on Matrix Analysis and Applications* 1994; 15(1): 228–272.

17. Hintz RM. Analytical methods in component modal synthesis. *AIAA Journal* 1975; 13(8): 1007–1016.

18. Hurty WC. Dynamic analysis of structural systems using component modes. *AIAA journal* 1965; 3(4): 678–685.

19. Kim JG, Lee PS. An enhanced Craig–Bampton method. *International Journal for Numerical Methods in Engineering* 2015; 103(2): 79–93.

20. Kim JH, Kim J, Lee PS. Improving the accuracy of the dual Craig-Bampton method. *Computers & Structures* 2017; 191: 22–32.

21. MacNeal RH. A hybrid method of component mode synthesis. *Computers & Structures* 1971; 1(4): 581–601.

22. Bampton MC, Craig Jr RR. Coupling of substructures for dynamic analyses.. *Aiaa Journal* 1968; 6(7): 1313–1319.

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

24. Bekas C, Saad Y. Computation of smallest eigenvalues using spectral Schur complements. *SIAM Journal on Scientific Computing* 2005; 27(2): 458–481.

25. Bennighof JK, Lehoucq RB. An automated multilevel substructuring method for eigenspace computation in linear elastodynamics. *SIAM Journal on Scientific Computing* 2004; 25(6): 2084–2106.

26. Elssel K, Voss H. An a priori bound for automated multilevel substructuring. *SIAM Journal on Matrix Analysis and Applications* 2006; 28(2): 386–397.

27. Gao W, Li XS, Yang C, Bai Z. An implementation and evaluation of the AMLS method for sparse eigenvalue problems. *ACM Transactions on Mathematical Software (TOMS)* 2008; 34(4): 1–28.

28. Hyun C, Boo SH, Lee PS. Improving the computational efficiency of the enhanced AMLS method. *Computers & Structures* 2020; 228: 106158.

29. Hyun C, Lee PS. A load balancing algorithm for the parallel automated multilevel substructuring method. *Computers & Structures* 2021; 257: 106649.

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

31. Bennighof J, Kaplan M, Muller M. Extending the frequency response capabilities of Automated Multi-Level Substructuring. In: ; 2000: 1574.

32. Bennighof J, Kaplan M, Muller M, Kim M. Meeting the NVH computational challenge: Automated Multi-Level Substructuring. In: Citeseer. ; 2000: 909–915.

33. Kropp A, Heiserer D. Efficient broadband vibro-acoustic analysis of passenger car bodies using an FE-based component mode synthesis approach. *J. Comput. Acoust.* 2003; 11(02): 139–157.

34. Ragnarsson P, Van Gaal T, Pluymers B, Donders S, Vandepitte D, Desmet W. Fast approximation of synthesized frequency response functions with automated multi-level substructuring (AMLS). *Finite elements in analysis and design* 2011; 47(2): 195–199.

35. Yin J, Voss H, Chen P. Improving eigenpairs of automated multilevel substructuring with subspace iterations. *Computers & Structures* 2013; 119: 115–124.

36. Kalantzis V. A Domain Decomposition Rayleigh–Ritz Algorithm for Symmetric Generalized Eigenvalue Problems. *SIAM Journal on Scientific Computing* 2020; 42(6): C410–C435.

37. Kalantzis V. A spectral Newton-Schur algorithm for the solution of symmetric generalized eigenvalue problems. *Electron. Trans. Numer. Anal.* 2020; 52: 132–153.

38. Kalantzis V, Kestyn J, Polizzi E, Saad Y. Domain decomposition approaches for accelerating contour integration eigenvalue solvers for symmetric eigenvalue problems. *Numerical Linear Algebra with Applications* 2018; 25(5): e2154.

39. Kalantzis V, Li R, Saad Y. Spectral Schur complement techniques for symmetric eigenvalue problems. *Electronic Transactions on Numerical Analysis* 2016; 45(LLNL-JRNL-691697).

40. Kalantzis V, Xi Y, Horesh L. Fast Randomized Non-Hermitian Eigensolvers Based on Rational Filtering and Matrix Partitioning. *SIAM Journal on Scientific Computing* 2021; 0(0): S791-S815. doi: 10.1137/20M1349217

41. Stathopoulos A, Saad Y, Fischer C. A Schur complement method for eigenvalue problems. In: ; 1995.

42. Kalantzis V, Xi Y, Saad Y. Beyond automated multilevel substructuring: Domain decomposition with rational filtering. *SIAM Journal on Scientific Computing* 2018; 40(4): C477–C502.

43. Yang C, Gao W, Bai Z, et al. An algebraic substructuring method for large-scale eigenvalue calculation. *SIAM Journal on Scientific Computing* 2005; 27(3): 873–892.

44. Anderson E, Bai Z, Bischof C, et al. *LAPACK Users' guide*. 9. SIAM . 1999.

45. Blackford LS, Choi J, Cleary A, et al. *ScaLAPACK users' guide*. SIAM . 1997.

46. Kalantzis V, Kollias G, Ubaru S, Nikolakopoulos AN, Horesh L, Clarkson K. Projection techniques to update the truncated SVD of evolving matrices with applications. In: PMLR. ; 2021: 5236–5246.

47. Go MS, Lim JH, Kim JG, Hwang Kr. A family of Craig–Bampton methods considering residual mode compensation. *Applied Mathematics and Computation* 2020; 369: 124822.

48. Hannukainen A, Malinen J, Ojalammi A. Efficient solution of symmetric eigenvalue problems from families of coupled systems. *SIAM Journal on Numerical Analysis* 2019; 57(4): 1789–1814.

49. Andrew A, Tan R. Computation of Derivatives of Repeated Eigenvalues and the Corresponding Eigenvectors of Symmetric Matrix Pencils. *SIAM Journal on Matrix Analysis and Applications* 1998; 20(1): 78-100. doi: 10.1137/S0895479896304332

50. Su Y, Lu T, Bai Z. 2D Eigenvalue Problems I: Existence and Number of Solutions. *arXiv preprint arXiv:1911.08109* 2019.

51. Cai D, Cai Z, Zhang S. Robust equilibrated error estimator for diffusion problems: mixed finite elements in two dimensions. *Journal of Scientific Computing* 2020; 83(1): 1–22.

52. Cai D, Cai Z. Hybrid a posteriori error estimators for conforming finite element approximations to stationary convection-diffusion-reaction equations. *arXiv preprint arXiv:2107.06341* 2021.

53. Cai D, Cai Z. A hybrid a posteriori error estimator for conforming finite element approximations. *Computer Methods in Applied Mechanics and Engineering* 2018; 339: 320–340.

54. Nakatsukasa Y. *Algorithms and Perturbation Theory for Matrix Eigenvalue Problems and the Singular Value Decomposition*. University of California, Davis . 2011.

55. Erlandson L, Cai D, Xi Y, Chow E. Accelerating parallel hierarchical matrix-vector products via data-driven sampling. In: IEEE. ; 2020: 749–758.

56. Cai D, Chow E, Erlandson L, Saad Y, Xi Y. SMASH: Structured matrix approximation by separation and hierarchy: Structured matrix approximation by separation and hierarchy. *Numerical Linear Algebra with Applications* 2018; 25(6).

57. Cai D, Nagy J, Xi Y. Fast and stable deterministic approximation of general symmetric kernel matrices in high dimensions. *arXiv e-prints* 2021: arXiv–2102.

58. Ruymbeek K, Meerbergen K, Michiels W. Calculating the minimal/maximal eigenvalue of symmetric parametrized matrices using projection. *arXiv preprint arXiv:1904.09923* 2019.

59. Gaul A, Schlömer N. Preconditioned recycling Krylov subspace methods for self-adjoint problems. *Electronic Transactions on Numerical Analysis* 2015; 44: 522–547.

60. Paige CC, Saunders MA. Solution of sparse indefinite systems of linear equations. *SIAM journal on numerical analysis* 1975; 12(4): 617–629.

61. Greenbaum A. *Iterative methods for solving linear systems*. SIAM . 1997.

□

APPENDIX

A A FORMULA FOR COMPUTING THE LEADING EIGENVECTOR DERIVATIVES

We now consider a computational procedure to approximate the leading derivatives of the eigenvectors $y_1, \dots, y_{n_{ev}}$. Differentiating both sides of the eigenvalue equation $S(\zeta)y_i(\zeta) + \theta_i(\zeta)dS(\zeta)y_i(\zeta) = 0$ with respect to the scalar $\zeta \notin \Lambda(B, M_B)$, and evaluating at the origin gives

$$\left(S + \theta_i M_s\right) dy_i = -\Big[(1 + d\theta_i)M_S + \theta_i d^2 S(0)\Big] y_i \equiv b^{(i)}, \tag{A1}$$

where

$$\frac{d^2 S(0)}{2} = E^T B^{-1} M_B B^{-1} M_B B^{-1} E + M_E^T B^{-1} M_E - M_E^T B^{-1} M_B B^{-1} E - E^T B^{-1} M_B B^{-1} M_E,$$

and

$$d\theta_i := d\theta_i = -1 + y_i^T d^2 S(0) y_i,$$

respectively. Solving the singular linear system in (A1) will determine dy_i only up to the eigenvector direction y_i. When θ_i is simple, the true eigenvector derivative can be computed by solving the augmented linear system, e.g., see[58],

$$\begin{bmatrix} S + \theta_i M_S & -M_S y_i \\ -y_i^T M_S & 0 \end{bmatrix} \begin{bmatrix} dy_i \\ d\theta_i \end{bmatrix} = \begin{bmatrix} (M_S + \theta_i d^2 S(0)) y_i \\ \frac{1}{2} y_i^T d^2 S(0) y_i \end{bmatrix},$$

leading to

$$dy_i = \left(\frac{1}{2} y_i^T d^2 S(0) y_i\right) y_i + \sum_{j=1,\ j\neq i}^{j=s} \left[y_j^T \frac{\left(M_S + \theta_j d^2 S(0)\right) y_i}{\theta_i - \theta_j} \right] y_j.$$

Nonetheless, the eigenvector direction y_i is already included in the Rayleigh-Ritz projection subspace, i.e., $y_i \in \texttt{range}\left(Y_{\{i,\tau\}}\right)$. Therefore, we can still focus on the solution of the singular linear system in (A1), and ignore the indeterminate direction y_i. The following discussion is a generalization of the approach described in[36].

Let $S = LL^T$ where $L \in \mathbb{R}^{s \times s}$ denotes a lower triangular matrix with real and positive diagonal entries. Then, we can re-write (A1) as

$$\left(I_s + \theta_i L^{-1} M_S L^{-T}\right)\left(L^T x\right) = L^{-1} b^{(i)}, \tag{A2}$$

where we replace dy_i with a generic vector x (since we no longer compute the exact dy_i). The linear system in (A2) is singular but consistent, and the eigenvalues of the matrix $I_s + \theta_i L^{-1} M_S L^{-T}$ are equal to $\frac{\theta_k - \theta_i}{\theta_k}$, with associated eigenvectors $L^T y_k,\ k = 1, \dots, s$. The solution in (A2) can be obtained by the MINRES iterative solver combined with deflation of the computed eigenvectors $y_1, \dots, y_{n_{ev}}$ [59,60]. In particular, the solution process can be split into two phases, as outlined in[36]. During the first phase we apply MINRES to the “deflated" linear system of equations

$$\mathcal{P}(I_s + \theta_i L^{-1} M_S L^{-T})\overline{x} = \mathcal{P} L^{-1} b^{(i)}, \tag{A3}$$

where $\mathcal{P} = I - W(W^T W)^{-1} W^T,\ K = L^T[y_1, \dots, y_{i-1}, y_{i+1}, \dots, y_{n_{ev}}]$, and $W = (I_s + \theta_i L^{-1} M_S L^{-T})K$. As soon as (A3) is solved, the solution of the original linear system is formed as

$$x = L^{-T}\left(\mathcal{Q}\bar{x} + (I - \mathcal{Q}) b^{(i)}\right), \tag{A4}$$

where $\mathcal{Q} = I - K\left(W^T W\right)^{-1} W^T$.

The effective condition number of the linear system in (A3) is equal to $\pi_i = \frac{\theta_{n_{ev}+1}}{\theta_s}\left(\frac{\theta_s - \theta_i}{\theta_{n_{ev}+1} - \theta_i}\right)$. This expression reveals that linear systems associated with smaller eigenvalues θ_i should converge faster, since their associated effective condition number is smaller[61].

Figure A1 illustrates the number of iterations required by deflated MINRES to approximate the eigenvector derivatives $dy_1, \dots, dy_{n_{ev}}$ of a finite difference discretization of the Laplace operator, up to a relative residual norm of 10^{-4}, where $p = \{4, 8, 16, 32\}$. As expected, linear systems associated with smaller eigenvalues θ_i converge faster. Moreover, the number of iterations is similar for all different values of p. In practice, it might be beneficial to deflate more than n_{ev} eigenvectors of the pencil $(S, -M_S)$, i.e., $n_{ev} + 10$, so as to speed-up convergence for those eigenvector derivatives dy_i associated with $i \approx n_{ev}$.

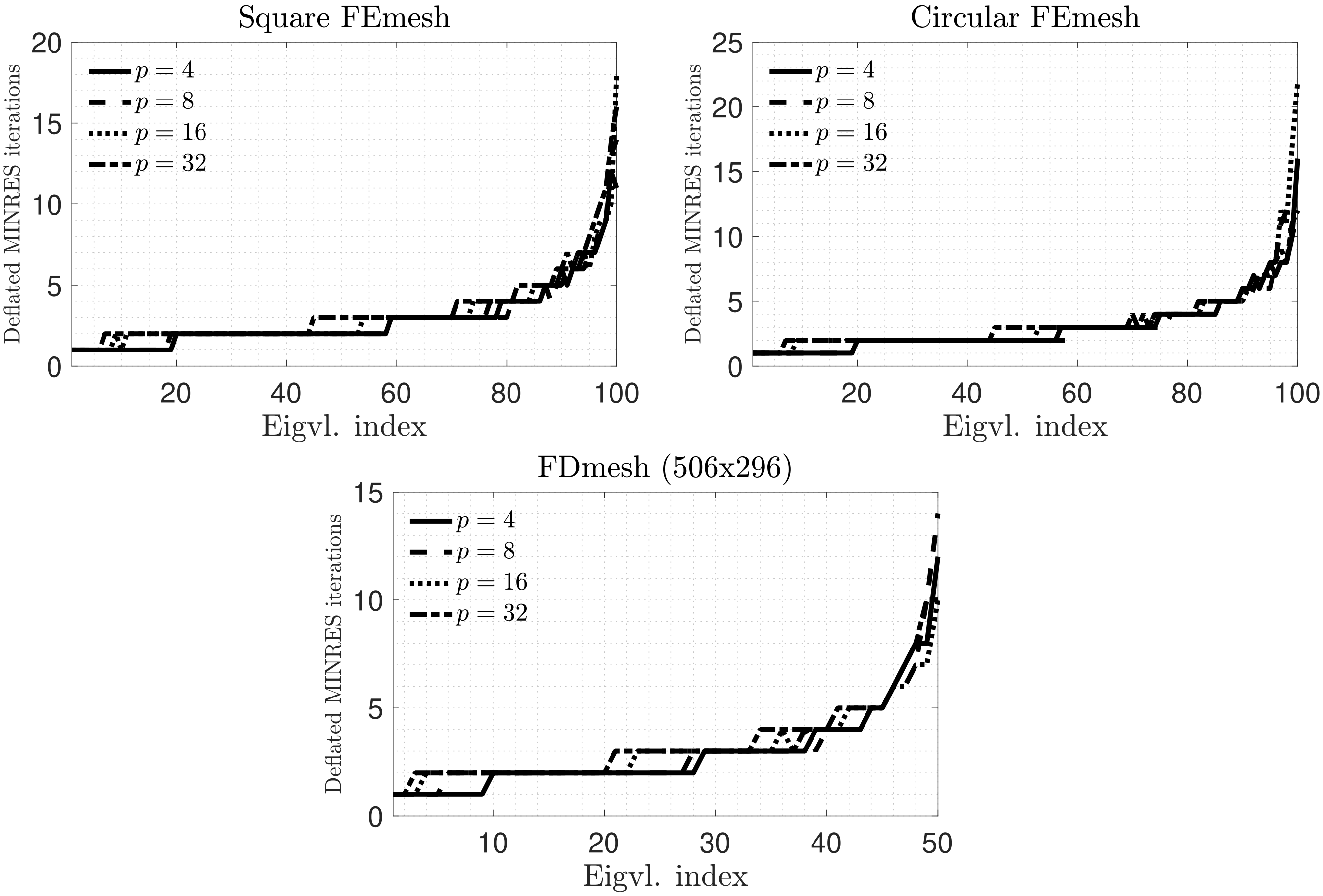

FIGURE A1 Number of iterations required by deflated preconditioned MINRES.