AN ADAPTIVE FAST SOLVER FOR A GENERAL CLASS OF
POSITIVE DEFINITE MATRICES VIA ENERGY DECOMPOSITION

THOMAS Y. HOU†, DE HUANG†, KA CHUN LAM‡, AND PENGCHUAN ZHANG§

Abstract. In this paper, we propose an adaptive fast solver for a general class of symmetric positive definite (SPD) matrices which include the well-known graph Laplacian. We achieve this by developing an adaptive operator compression scheme and a multiresolution matrix factorization algorithm which achieve nearly optimal performance on both complexity and well-posedness. To develop our adaptive operator compression and multiresolution matrix factorization methods, we first introduce a novel notion of energy decomposition for SPD matrix $A$ using the representation of energy elements. The interaction between these energy elements depicts the underlying topological structure of the operator. This concept of decomposition naturally reflects the hidden geometric structure of the operator which inherits the localities of the structure. By utilizing the intrinsic geometric information under this energy framework, we propose a systematic operator compression scheme for the inverse operator $A^{-1}$. In particular, with an appropriate partition of the underlying geometric structure, we can construct localized basis by using the concept of interior and closed energy. Meanwhile, two important localized quantities are introduced, namely, the error factor and the condition factor. Our error analysis results show that these two factors will be the guidelines for finding the appropriate partition of the basis functions such that prescribed compression error and acceptable condition number can be achieved. By virtue of this insight, we propose the patch pairing algorithm to realize our energy partition framework for operator compression with controllable compression error and condition number.

Key words. energy decomposition, graph Laplacian, SPD matrix, fast solver, operator compression, multiresolution matrix decomposition

AMS subject classifications. 15A09, 65F08, 68R10

DOI. 10.1137/17M1140686

1. Introduction. Fast algorithms for solving symmetric positive definite (SPD) linear systems have found broad applications across both theory and practice, including machine learning [5, 9, 31], computer vision [4, 41, 6], image processing [1, 10, 24], computational biology [13, 25], etc. For instance, the graph Laplacian, which has deep connection between the combinatorial properties of the graph $G$ and the linear algebraic properties of the Laplacian $L$, is one of the foundational problems in data analysis. Performing finite element simulation of a wide range of physical systems will also introduce the corresponding stiffness matrix, which is also symmetric and positive definite.

The philosophy of this work is inspired by the spectral graph theory [26, 12], in which the spectrum and the geometry of graphs are highly correlated. By computing the spectrum of the graph, the intrinsic geometric information can be directly obtained, and various applications can be found [7, 18]. However, the price for finding the spectrum of graph is relatively expensive, as it involves solving global
eigenproblems. On the contrary, the algebraic multigrid (AMG) method is a purely matrix-based multiresolution-type solver. It simply uses the interaction of nonzero entries within the matrix as an indicator to describe the geometry implicitly. These stimulating techniques and concepts motivate us to look into the problems from two different points of view and search for a brand new framework which can integrate the advantages from both ends.

In this paper, we propose an adaptive fast solver for a general class of SPD matrices. We achieve this by developing an adaptive operator compression scheme and a multiresolution matrix factorization algorithm both with nearly optimal performance on complexity and well-posedness. These methods are developed based on a newly introduced framework, namely, the energy decomposition for SPD matrix $A$ to extract its hidden geometric information. For the ease of discussion, we first consider $A = L$, the graph Laplacian of an undirected graph $G$. Under this framework, we reformulate the connectivity of subgraphs in $G$ as the interaction between energies. These interactions reveal the intrinsic geometric information hidden in $L$. In particular, this framework naturally leads into two important local measurements, which are the error factor and the condition factor. Computing these two measurements only involves solving a localized eigenvalue problem, and consequently no global computation or information is involved. These two measurements serve as guidance to define an appropriate partition of the graph $G$. Using this partition, a modified coarse space and corresponding basis with exponential decaying property can be constructed. Compression of $L^{-1}$ can thus be achieved. Furthermore, the systematic clustering procedure of the graph regardless of the knowledge of the geometric information allows us to introduce a multiresolution matrix decomposition (MMD) framework for graph Laplacian and, more generally, SPD linear systems. In particular, following the work in [27], we propose a nearly linear time fast solver for general SPD matrices. Given the prescribed well-posedness requirement (i.e., the condition factor), every component from MMD will be a well-conditioned, lower-dimensional SPD linear system. Any generic iterative solver can then be applied in parallel to obtain the approximated solution of the given arbitrary SPD matrix satisfying the prescribed accuracy.

1.1. Overview of our results. Given an $n \times n$ SPD matrix $A$ with $m$ nonzero entries, our ultimate goal is to develop a fast algorithm to efficiently solve $Ax = b$ or, equivalently, compress the solver $A^{-1}$ with desired compression error. We make the following assumptions on the matrix $A$. First, $\lambda_{\min}(A) = O(1)$ for well-posedness, where $\lambda_{\min}(A)$ is the minimum eigenvalue of the matrix $A$. Second, the spectrum of the matrix $A$ is broad-banded. Third, it is stemmed from summation of symmetric and positive semidefinite (SPSD) matrices. We remark that the second assumption can be interpreted as the sparsity requirement of $A$, which is the existence of some intrinsic, localized geometric information. For instance, if $A = L$ is a graph Laplacian and $A$ is the corresponding matrix, such sparsity can be described by the requirement

$$\#N_k(i) = O(k^d) \forall i,$$

where $\#N_k(i)$ is the number of vertices near the vertex $i$ with logic distance smaller than $k$ (i.e., number of nonzero off-diagonal entries on row $i$ of $A^k$) and $d$ is the geometric dimension of the graph (i.e., the optimal embedding dimension). This is equivalent to assuming that the portion of long interaction edges is small. The third assumption, in many concerning cases, is a natural consequence during the assembling of the matrix $A$. In particular, a graph Laplacian $L$ can be viewed as a summation
of 2-by-2 matrices representing edges in the graph $G$. These 2-by-2 matrices are SPSD matrices and can be obtained automatically if $G$ is given. Another illustrative example is the patchwise stiffness matrix of a finite element from the discretization of PDEs using FE-type methods.

To compress the solver $A^{-1}$ (where $A$ satisfies the above assumptions) with a desired error bound, we adopt the idea of constructing modified coarse space as proposed in [22, 27, 16]. The procedure is summarized in Figure 1. Note that one common strategy of these PDE approaches is to make use of the natural partition under some a priori geometric assumption on the computational domain. In contrast, we adaptively construct an appropriate partition using the energy decomposition framework, which requires no a priori knowledge related to the underlying geometry of the domain. This partitioning idea is prerequisite and advantageous in the scenario when no explicit geometry information is provided, especially in the case of graph Laplacian. Therefore, one of our main contributions is to develop various criteria and systematic procedures to obtain an appropriate partition $P = \{P_j\}_{j=1}^M$ (i.e., graph partitioning in the case of graph Laplacian) which reveals the structural property of $A$.

Leaving aside the difficulties of finding an appropriate partition, our next task is to define a coarse space $\Phi$ such that $\|x - P_\Phi x\|_2 \leq \epsilon \|x\|_A$. As shown in [27, 16], having such requirement, together with the modification of coarse space $\Phi$ into $\Psi = A^{-1}(\Phi)$, we have $\|A^{-1} - P_\Psi A^{-1}\|_2 \leq \epsilon^2$. This affirms that $\Phi$ must be constructed carefully in order to achieve the prescribed compression accuracy. Further, under the energy decomposition setting, we can ensure such requirement by simply considering local accuracy $\|x - P_\Phi x\|_2$, which in turn gives a local computation procedure for checking the qualification of the local $\Phi_j$. Specifically, we introduce the error factor $\epsilon(P, q) = \max_{P_j \in P} \frac{1}{\sqrt{\lambda_{q+1}(P_j)}}$ (where $\lambda_{q+1}(P_j)$ corresponds to the $(q+1)$th smallest eigenvalue of some eigenproblem defined on the patch $P_j$) as our defender to strictly control the overall compression error. The error factor guides us to construct a subspace $\Phi_j^q \in \text{span}(P_j)$ for every patch $P_j$ satisfying the local accuracy and, eventually, the global accuracy requirement. Afterwards, we apply the formulation of the modified coarse space $\Psi$ to build up the exponential decaying basis $\psi_j$ for the operator compression. To be general, we reformulate the coarse space modification procedure by using purely matrix-based arguments. In addition, this reformulation immediately introduces another criterion, called the condition factor $\delta(P_j)$, over every patch $P_j \in P$. This second measurement serves as another defender to control the well-posedness of our compressed operator. Similar to the error factor, the condition factor is a local measurement which can be obtained by solving a partial local eigenproblem. This local restriction can naturally convey in a global sense to bound the maximum eigenvalue of the compressed operator. In particular, we prove that the compressed operator $A_{st}$ satisfies $\kappa(A_{st}) \leq \max_{P_j \in P} \delta(P_j) \|A^{-1}\|_2$.

Up to this point, we can see that the choice of the partition has a direct influence on both the accuracy and the well-posedness of the compressed operator. In this work, we propose a nearly linear time partitioning algorithm which is purely matrix-based and with complexity $O(d \cdot s^2 \cdot \log s \cdot n) + O(\log s \cdot n \cdot \log n)$, where $s$ is the average
patch size and \( d \) is the intrinsic geometric dimension. With the relationship between the **error factor** and **condition factor**, we can reversely treat the local properties as the blueprint to govern the process of partitioning. This in turn regularizes the condition number of the compressed operator such that computational complexity of solving the original linear system can be magnificently reduced.

Having a generic operator compression scheme, we follow the idea in [27] to extend the compression scheme hierarchically to form an MMD algorithm. However, instead of using a precedent nested partitioning of the domain, we perform the decomposition level by level in a recursive manner. In other words, every new level is generated inductively from the previous level (subject to some uniform well-posedness constraints) by applying our adaptive partitioning technique. This provides more flexibility and convenience to deal with various and even unknown multiresolution behavior appearing in the matrix. This decomposition further leads us to develop a fast solver for SPD matrices with time complexity \( O(m \log n \cdot (\log \epsilon^{-1} + \log n)^c \log \epsilon^{-1}) \), where \( m \) is the number of nonzero entries of \( A \) and \( c \) is some absolute constant depending only on the geometric property of \( A \). We would like to emphasize that the construction of the appropriate partition is essential to the formation of the hierarchical decomposition procedure. Principally, the hidden geometric information of \( A \) can be subtly recovered from the inherited energy decomposition of compressed operator using our framework. The interaction between these inherited energies serve similar purposes as the energy elements of \( A \). Therefore, we can recognize the compressed operator as an initial operator in the second level of the decomposition procedure and proceed to the next level repeatedly. We also remark that with our right choice of partitioning, the sparsity and the well-posedness properties of the compressed operator can be inherited among layers. This nice property enables us to decompose the original problem of solving \( A^{-1} \) into sets of independent problems with similar complexity and condition, which favors the parallel implementation of the solver.

1.2. Previous works. Recently, several works relevant to the compression of elliptic operators with heterogeneous and highly varying coefficients have been proposed. Målqvist and Peterseim et al. [22, 20] construct localized multiscale basis functions from the modified coarse space \( V_H^{ms} = V_H - \mathcal{F}V_H \), where \( V_H \) is the original coarse space spanned by conforming nodal basis and \( \mathcal{F} \) is the energy projection onto the space \( (V_H)^\perp \). The exponential decaying property of these modified basis has also been shown both theoretically and numerically. Meanwhile, a beautifully insightful work from Owhadi [27] reformulates the problem from the perspective of decision theory using the idea of *gamblets* as the modified basis. In particular, a coarse space \( \Phi \) of measurement functions is constructed from Bayesian perspective, and the gamblet space is explicitly given as \( \Psi = A^{-1}(\Phi) \), which turns out to be a counterpart of the modified coarse space in [22]. In addition, the basis of \( \Phi \) is generalized to nonconforming measurement functions, and the gamblets are still proven to decay exponentially such that localized computation is made possible. Hou and Zhang in [16] extend these works such that localized basis functions can also be constructed for higher-order strongly elliptic operators. Owhadi further generalizes these frameworks to a more unified methodology for arbitrary elliptic operators on Sobolev spaces in [28] using the Gaussian process interpretation. Note that for the above-mentioned works, since the problems they considered originated from PDE-type modeling, the computational domains are naturally assumed to be given; that is, the partition \( \mathcal{P} \) can be obtained directly (which is not available for graph Laplacians or general SPD
matrices). This assumption greatly helps the immersion of the nested coarse spaces with different scales into the computational domain. In other words, the exponential decaying property of the basis can be precisely achieved.

Recall that for solving linear systems exhibiting multiple scales of behavior, the class of multiresolution methods decomposes the problem additively in terms of different scales of resolution. This captures the features of different scales and allows us to treat these components differently. For instance, the enlightening geometric multigrid (GMG) methods [11, 40, 30] provide fast solvers for linear systems which are stemmed from discretization of linear elliptic differential equations. The main idea is to accelerate the convergence of basic iterative methods by introducing a nested structure on the computational domain so that successive subspace correction can be performed. However, the performance is hindered when the regularity of the coefficients is lost. To overcome this deficiency, an enormous amount of significant progress has been achieved. Numerous methods ranging from geometry-specific to purely algebraic/matrix-based approaches have been developed (see [37, 2, 15] for review). Using the tools of compressing the operator $A^{-1}$ possessing multiple scale features, Owhadi in [27] also proposes a straightforward but intelligible way to solve the roughness issue. By introducing a natural nested structure on the given computational domain, a systematic multiresolution algorithm for hierarchically decomposing elliptic operators is proposed. This in turn derives a nearly linear complexity solver with guaranteed prescribed error bounds. The efficiency of this multilevel solver is guaranteed by carefully choosing a nested structure of measurement functions $\Phi$, which satisfies (i) the Poincaré inequality, (ii) the inverse Poincaré inequality, and (iii) the frame inequality. In [28], Owhadi and Scovel extend the result to problems with general SPD matrices, where the existence of $\Phi$ satisfying (i), (ii), and (iii) is assumed. In particular, for discretization of continuous linear bijections from $H^s(\Omega)$ to $H^{-s}(\Omega)$ or $L^2(\Omega)$ space, these assumptions are shown to hold true using prior information on the geometry of the computational domain $\Omega$. However, the practical construction of this nested global structure $\Phi$ is an essentially hard problem when no intrinsic geometric information is provided a priori. To solve this generic problem, we introduce the energy decomposition and the inherited system of energy elements. Instead of a priori assuming the existence of such nested structure $\Phi$, we use the idea of inherited energy decomposition to levelwisely construct $\Phi$ and the corresponding energy decomposition by using local spectral information and an adaptive clustering technique.

On the other hand, to mimic the functionality and convergence behavior of GMG without providing the nested meshes, the AMG methods [37, 39, 38] speculate the coefficients through the connectivity information in the given matrix to define intergrid transfer operators, which avoids the direct construction of the restriction and relaxation operators in GMG methods. Intuitively, the connectivity information discloses the hidden geometry of the problem subtly. This purely algebraic framework bypasses the “geometric” requirement in GMG and is widely used in practice on graphs with sufficiently nice topologies. In particular, a recent AMG method called $LAMG$ has been proposed by Livne and Brandt [21], where the run time and storage of the algorithm are empirically demonstrated to scale linearly with the number of edges. We would like to emphasize that the difference between our proposed solver and a general recursive-type iterative solver is the absence of nested iterations. Our solver decomposes the matrix adaptively according to the inherited multiple scales of the matrix itself. The matrix decomposition divides the original problem into components of
controllable well-conditioned, lower-dimensional SPD linear systems, which can then be solved in parallel using any generic iterative solver. In other words, this decomposition also provides a parallelizable framework for solving SPD linear systems.

Another inspiring stream of nearly linear time algorithm for solving graph Laplacian system was given by Spielman and Teng [32, 33, 34, 35, 36]. With the innovative discoveries in spectral graph theory and graph algorithms, such as the fast construction low-stretch spanning trees and clustering scheme, they successfully employ all these important techniques in developing an effective preconditioned iterative solver. Later, Koutis, Miller, and Peng [19] follow these ideas and simplify the solver with computation complexity $O(m \log n \log n \log \epsilon^{-1})$, where $m$ and $n$ are the number of edges and vertices, respectively. In contrast, we employ the idea of modified coarse space to compress a general SPD matrix (i.e., the graph Laplacian in this case) hierarchically with the control of sparsity and well-posedness.

1.3. Outline. In subsection 2.1, we will introduce the foundation of our work, which is the notion of energy decomposition of general SPD matrices. Section 3 discusses the construction of the coarse space and its corresponding modified coarse space, which serves to construct the basis with exponential decaying property. Concurrently, the local measurements error factor and the condition factor are introduced. The analysis in this section will guide us to design the systematic algorithm for constructing the partition $\mathcal{P}$, which is described in section 4. Discussion of the computational complexity is also included. To demonstrate the efficacy of our partitioning algorithm, two numerical results are reported in section 5. Furthermore, we extend the idea of operator compression into MMD in section 6. In the meantime, we propose the concept of localization of MMD and the inherited locality of the compressed operator. These essential ingredients guide us to develop the parallelizable solver with nearly linear time complexity. Error estimate and numerical results are reported to show the efficacy of this proposed algorithm. A conclusion and a discussion of future works are included in section 7. For better readability, most of the proofs are moved to the appendix.

2. Preliminaries. In this paper we aim to solve the linear system $Lx = b$, where $L$ is the Laplacian of a undirected, positive-weighted graph $G = \{V; E, W\}$, i.e.,

\begin{equation}
L_{ij} = \begin{cases}
\sum_{(i,j') \in E} w_{ij'} & \text{if } i = j; \\
-w_{ij} & \text{if } i \neq j \text{ and } (i,j) \in E; \\
0 & \text{otherwise}.
\end{cases}
\end{equation}

We allow for the existence of self-loops $(i, i) \in E$. When $L$ is singular, we mean to solve $x = L^\dagger b$, where $L^\dagger$ is the pseudoinverse of $L$. Our algorithm will be based on a fast clustering technique using local spectral information to give a good partition of the graph, upon which special local basis will be constructed and used to compress the operator $L^{-1}$ into a low-dimensional approximation $L^{-1}_{\text{com}}$ subject to a prescribed accuracy.

As we will see, our clustering technique exploits local pathwise information of the graph $G$ by operating on each single edge in $E$, which can be easily adapted to a larger class of linear systems with symmetric, positive semidefinite matrix. Notice that the contribution of an edge $(i,j) \in E$ with weight $w_{ij}$ to the Laplacian matrix $L$ is simply
and we have \( L = \sum_{(i,j) \in E} E_{ij} \). In view of such matrix decomposition, our algorithm works for any symmetric, positive semidefinite matrix \( A \) that has a similar decomposition \( A = \sum_{k=1}^{m} E_{k} \) with each \( E_{k} \succeq 0 \). Therefore, we will theoretically develop our method for general decomposable SPD matrices. Also, we assume that \( A \) is invertible, as we can easily generalize our method to the case when \( A^T b \) is pursued.

### 2.1. Energy decomposition

In this section, we will introduce the idea of energy decomposition and the corresponding mathematical formulation which motivates the methodology for solving linear systems with energy-decomposable linear operator. Let \( A \) be a \( n \times n \) SPD matrix. We define the **energy decomposition** as follows.

**Definition 2.1** (energy decomposition). We call \( \{ E_{k} \}_{k=1}^{m} \) an energy decomposition of \( A \) and \( E_{k} \) to be an energy element of \( A \) if

\[
A = \sum_{k=1}^{m} E_{k}, \quad E_{k} \succeq 0 \quad \forall k = 1, \ldots, m,
\]

where \( E_{k} \succeq 0 \) means \( E_{k} \) is positive semidefinite. Intuitively, the underlying structural (geometric) information of the original matrix \( A \) can be realized through an appropriate energy decomposition. And to preserve as much detailed information of \( A \) as possible, it is better to use the finest energy decomposition that we can have, which actually comes naturally from the generating of \( A \), as we will see in some coming examples. More precisely, for an energy decomposition \( \mathcal{E} = \{ E_{k} \}_{k=1}^{m} \) of \( A \), if there is some \( E_{k} \) that has its own energy decomposition \( E_{k} = E_{k,1} + E_{k,2} \) that comes naturally, then the finer energy decomposition \( \mathcal{E}_{\text{fine}} = \{ E_{k} \}_{k=2}^{m} \cup \{ E_{k,1}, E_{k,2} \} \) is more preferred, as it gives us more detailed information of \( A \). However, one would see that any \( E_{k} \) can have some trivial decomposition \( E_{k} = \frac{1}{2} E_{k} + \frac{1}{2} E_{k} \), which makes no essential difference. To make it clear what should be the finest underlying energy decomposition of \( A \) that we will use in our algorithm, we first introduce the neighboring relation between energy elements and basis.

Let \( \mathcal{E} = \{ E_{k} \}_{k=1}^{m} \) be an energy decomposition of \( A \) and \( \mathcal{V} = \{ v_{i} \}_{i=1}^{n} \) be an orthonormal basis of \( \mathbb{R}^{n} \). We introduce the following notation:

- For any \( E \in \mathcal{E} \) and any \( v \in \mathcal{V} \), we denote \( E \sim v \) if \( v^{T} E v > 0 \) (or equivalently \( Ev \neq 0 \), since \( E \succeq 0 \)).
- For any \( u, v \in \mathcal{V} \), we denote \( u \sim v \) if \( u^{T} A v \neq 0 \) (or equivalently \( \exists E \in \mathcal{E} \) such that \( u^{T} E v \neq 0 \)).

As an immediate example, if we take \( \mathcal{V} \) to be the set of all distinct eigenvectors of \( A \), then \( v \not\sim u \) for any two \( v, u \in \mathcal{V} \); namely, all basis functions are isolated, and everything is clear. But such choice of \( \mathcal{V} \) is not trivial in that we know everything about \( A \) if we know its eigenvectors. Therefore, instead of doing things in the frequency space, we assume the least knowledge of \( A \) and work in the physical space; that is, we will choose \( \mathcal{V} \) to be the natural basis \( \{ e_{i} \}_{i=1}^{n} \) of \( \mathbb{R}^{n} \) in all practical use. But for theoretical analysis, we still use the general basis notation \( \mathcal{V} = \{ v_{i} \}_{i=1}^{n} \).
Also, for those who are familiar with graph theory, it is more convenient to understand the sets $V, E$ from a graph perspective. Indeed, one can keep in mind that $G = \{V, E\}$ is the generalized concept of undirected graphs, where $V$ stands for the set of vertices and $E$ stands for the set of edges. For any vertices (basis) $v, u \in V$, and any edge (energy) $E \in E$, $v \sim E$ means that $E$ is an edge of $v$, and $v \sim u$ means that $v$ and $u$ share some common edge. However, different from the traditional graph setting, here one edge (energy) $E$ may involve multiple vertices instead of just two, and two vertices (basis) $v, u$ may share multiple edges that involve different sets of vertices. Further, the spectrum magnitude of the “multivertex edge” $E$ can be viewed as a counterpart of edge weight in a graph setting. Conversely, if the problem comes directly from a weighted graph, then one can naturally construct the sets $V$ and $E$ from the vertices and edges of the graph, as we will see in Example 2.10.

**Definition 2.2 (neighboring).** Let $\mathcal{E} = \{E^k\}_{k=1}^m$ be an energy decomposition of $A$ and $V = \{v_i\}_{i=1}^n$ be an orthonormal basis of $\mathbb{R}^n$. For any $E \in \mathcal{E}$, we define $N(E; V) := \{v \in V : E \sim v\}$ to be the set of $v \in V$ neighboring $E$. Similarly, for any $v \in V$, we define $N(v; \mathcal{E}) := \{E \in \mathcal{E} : E \sim v\}$ and $N(v) := \{u \in V : u \sim v\}$ to be the set of $E \in \mathcal{E}$ and the set of $u \in V$ neighboring $v \in V$, respectively. Furthermore, for any $S \subset V$ and any $E \in \mathcal{E}$, we denote $E \sim S$ if $N(E; V) \cap S \neq \emptyset$ and $E \in S$ if $N(E; V) \subset S$.

In what follows, we will see that if two energy elements $E_k, E'_k$ have the same neighbor basis, namely, $N(E_k; V) = N(E'_k; V)$, then there is no need to distinguish between them since it is the neighboring relation between energy elements and basis that matters in how we make use of the energy decomposition. Therefore, we say an energy decomposition $\mathcal{E} = \{E^k\}_{k=1}^m$ is the finest underlying energy decomposition of $A$ if no $E_k \in \mathcal{E}$ can be further decomposed as

$$E_k = E_{k,1} + E_{k,2},$$

where either $N(E_{k,1}; V) \subset N(E_k; V)$ or $N(E_{k,2}; V) \subset N(E_k; V)$. From now on, we will always assume that $\mathcal{E} = \{E^k\}_{k=1}^m$ is the finest underlying energy decomposition of $A$ that comes along with $A$.

Using the neighboring concept between energy elements and orthonormal basis, we can then define various energies of a subset $S \subset V$ as follows.

**Definition 2.3 (restricted, interior, and closed energy).** Let $\mathcal{E} = \{E^k\}_{k=1}^m$ be an energy decomposition of $A$. Let $S$ be a subset of $V$ and $P_S$ be the orthogonal projection onto $S$. The **restricted energy** of $S$ with respect to $A$ is defined as

$$A_S := P_S A P_S,$$

the **interior energy** of $S$ with respect to $A$ and $\mathcal{E}$ is defined as

$$A^\mathcal{E}_S = \sum_{E \in S} E,$$

and the **closed energy** of $S$ with respect to $A$ and $\mathcal{E}$ is defined as

$$\overline{A}^\mathcal{E}_S = \sum_{E \in S} E + \sum_{E \not\in S, E \sim S} P_S E^d P_S,$$

where

$$E^d = \sum_{v \in V} \left( \sum_{u \in V} |v^T E u| \right) v u^T = \sum_{v \sim E} \left( \sum_{u \sim v} |v^T E u| \right) v u^T.$$
is called the \textit{diagonal concentration} of $E$, and we have

\begin{equation}
P_S E^d P_S = \sum_{v \in S, v \sim E} \left( \sum_{u \sim v} |v^T E u| \right) v v^T.
\end{equation}

Remark 2.4. The \textit{restricted energy} of $S$ can be simply viewed as the restriction of $A$ on the subset $S$. The \textit{interior energy (closed energy)} of $S$ is $A_S$ excluding (including) contributions from other energy elements $E \notin S$ neighboring $S$. The following example illustrates the idea of various energies introduced in Definition 2.3 by considering the one-dimensional discrete Laplace operator with Dirichlet boundary conditions.

\textbf{Example 2.5.} Consider $A$ to be the $(n+1) \times (n+1)$ tridiagonal matrix with entries $-1$ and $2$ on off-diagonals and diagonals, respectively. Let

\begin{align}
E_1 &= \begin{pmatrix} -1 & 1 \\ -1 & 0 \end{pmatrix}, \\
E_n &= \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}, \\
E_k &= \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}
\end{align}

for $k = 2, \ldots, n-1$. Let $V = \{ e_i \}_{i=0}^n$ be the standard orthonormal basis for Euclidean space $\mathbb{R}^{n+1}$. Formally, $E_k$ is the edge between $e_k-1$ and $e_k$. If $S = \{ e_3, e_4, e_5, e_6 \}$, then we have

\begin{align}
A_S &= \begin{pmatrix} 0 & 1 & -1 & 0 \\ -1 & 2 & -1 & 0 \\ -1 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \\
\Delta_S &= \begin{pmatrix} 0 & -1 & -1 & 0 \\ -1 & 2 & -1 & 0 \\ -1 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \\
\overline{A}_S &= \begin{pmatrix} 0 & -1 & -1 & 0 \\ -1 & 2 & -1 & 0 \\ -1 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 \end{pmatrix}
\end{align}

Recall that the \textit{interior energy} $A_S^e = \sum_{E_k \in S} E_k = \sum_{k=4}^6 E_k$, while the \textit{closed energy}

\begin{align}
\overline{A}_S &= A_S^e + \sum_{E \notin S, E \sim S} P_S E^d P_S \\
&= A_S^e + |e_3^T E_3 e_2| e_2 e_3^T + |e_3^T E_3 e_4| e_4 e_3^T + |e_5^T E_7 e_6| e_6 e_5^T + |e_6^T E_7 e_6| e_6 e_6^T
\end{align}

includes the partial contributions from other energy elements $E \notin S$ neighboring $S$, which are $E_3$ and $E_7$, respectively.

Remark 2.6.

- Notice that any eigenvector $x$ of $A_S$ (or $A_S^e$, $\overline{A}_S$) corresponding to nonzero eigenvalue must satisfy $x \in \text{span}(S)$. In this sense, we also say $A_S$ (or $A_S^e$, $\overline{A}_S$) is local to $S$.
- For any energy $E$, we have $E \preceq E^d$ since for any $x = \sum_{i=1}^n c_i v_i$, we have

\begin{align}
x^T E x &= \sum_{i} c_i^2 v_i^T E v_i + \sum_{i \neq j} 2 c_i c_j v_i^T E v_j \\
&\leq \sum_{i} c_i^2 v_i^T E v_i + \sum_{i \neq j} (c_i^2 + c_j^2) |v_i^T E v_j| = \sum_{i} \sum_{j} c_i^2 |v_i^T E v_j| = x^T E^d x.
\end{align}

\textbf{Proposition 2.7.} For any $S \subset V$, we have that $A_S^e \preceq A_S \preceq \overline{A}_S$. 
Proof. We have

\[ A_s^\mathcal{E} = \sum_{E \in \mathcal{S}} E \leq \sum_{E \in \mathcal{S}} E + \sum_{E \notin \mathcal{S}, E \sim S} E \leq \sum_{E \in \mathcal{S}} E + \sum_{E \notin \mathcal{S}, E \sim S} P_s E^d P_s = A_S^\mathcal{E}. \]

Notice that \( P_s E \mathcal{P}_S = E \) for \( E \in \mathcal{S} \) and \( P_s E \mathcal{P}_S = 0 \) for \( E \notin \mathcal{S} \); thus,

\[ A_S = P_s A \mathcal{P}_S = \sum_{E \in \mathcal{S}} E + \sum_{E \notin \mathcal{S}, E \sim S} P_s E \mathcal{P}_S, \]

and the desired result follows.

**Definition 2.8 (partition of basis).** Let \( \mathcal{V} = \{ v_i \}_{i=1}^n \) be an orthonormal basis of \( \mathbb{R}^n \). We say \( \mathcal{P} = \{ P_j \}_{j=1}^M \) is a *partition* of \( \mathcal{V} = \{ v_i \}_{i=1}^n \) if (i) \( P_j \subset \mathcal{V} \forall j \), (ii) \( P_j \cap P_{j'} = \emptyset \) if \( j \neq j' \), and (iii) \( \bigcup_{j=1}^M P_j = \mathcal{V} \).

Again one can see the partition of basis as partition of vertices. This partition \( \mathcal{P} \) is the key to construction of local basis for operator compression purpose. The following proposition serves to bound the matrix \( A \) from both sides with blocked (patched) matrices, which will further serve to characterize properties of local basis.

**Proposition 2.9.** Let \( \mathcal{E} = \{ E_k \}_{k=1}^n \) be an energy decomposition of \( A \) and \( \mathcal{P} = \{ P_j \}_{j=1}^M \) a partition of \( \mathcal{V} \). Then

\[
\sum_{j=1}^M A_{P_j}^\mathcal{E} \preceq A \preceq \sum_{j=1}^M A_{P_j}^\mathcal{E}.
\]

**Proof.** Let \( \mathcal{E}_\mathcal{P} = \{ E \in \mathcal{E} : \exists P_j \in \mathcal{P} \text{ such that } E \in P_j \} \) and \( \mathcal{E}_\mathcal{P}^c = \mathcal{E} \setminus \mathcal{E}_\mathcal{P} \). Recall that \( E \in P_j \) if \( N(E, \mathcal{V}) \subset P_j \) (see Definition 2.2). We will use \( P_j \) to denote the orthogonal projection onto \( P_j \). Since \( P_j \cap P_{j'} = \emptyset \) for \( j \neq j' \), we have \( \sum_j P_j = \text{Id} \).

Then

\[
\sum_{j} A_{P_j}^\mathcal{E} = \sum_{j} \sum_{E \in P_j} E \leq \sum_{j} \sum_{E \in P_j} E + \sum_{E \notin \mathcal{E}_\mathcal{P}} E \\
\leq \sum_{j} \sum_{E \in P_j} E + \sum_{E \notin \mathcal{E}_\mathcal{P}} \left[ \left( \sum_{j} P_j \right) E^d \left( \sum_{j'} P_{j'} \right) \right] \\
= \sum_{j} \sum_{E \in P_j} E + \sum_{E \notin \mathcal{E}_\mathcal{P}} \left( \sum_{j} P_j E^d P_j \right) \\
= \sum_{j} \left( \sum_{E \in P_j} E + \sum_{E \notin P_j, E \sim P_j} P_j E^d P_j \right) = \sum_{j} A_{P_j}^\mathcal{E}.
\]

We have used the fact that \( P_j E^d P_{j'} = 0 \) for \( j \neq j' \). Notice that

\[
A = \sum_{E \in \mathcal{E}_\mathcal{P}} E + \sum_{E \notin \mathcal{E}_\mathcal{P}} E = \sum_{j} \sum_{E \in P_j} E + \sum_{E \notin \mathcal{E}_\mathcal{P}} E,
\]

and the desired result follows.
Throughout the paper, we will always assume that $A$ has a finest energy decomposition $E = \{E_k\}_{k=1}^{m}$ and all the other discussed energies of $A$ are constructed from $E$ with respect to some orthonormal basis $V$ (by taking interior or closed energy). Therefore, we will simply use $A_{S}, A_{\bar{S}}$ to denote $A_{E_{S}}, A_{E_{\bar{S}}}$ for any $S \subset V$.

**Example 2.10.** Consider $L$ to be the graph Laplacian matrix of the graph given in Figure 2. For graph Laplacian, an intrinsic energy decomposition arises during the assembling of the matrix in which the energy element is defined over each edge (see (2)). Now suppose we have given the partition $P = \{P_j\}_{j=1}^{3}$ with $P_1 = [e_1, e_2, e_3]$, $P_2 = [e_4, e_5, e_6, e_7]$, and $P_3 = [e_8, e_9, e_{10}, e_{11}]$, where $e_i$ are the standard basis of $\mathbb{R}^{11}$.

Then we can obtain $L_{P_j}$ and $\overline{L}_{P_j}$ as follows:

$$L_{P_1} = \begin{pmatrix} 4 & -2 & -2 \\ -2 & 4 & -2 \\ -2 & -2 & 4 \end{pmatrix} \quad L_{P_2} = \begin{pmatrix} 5 & -2 & -1 \\ -2 & 4 & -2 \\ -1 & -2 & 5 \end{pmatrix} \quad L_{P_3} = \begin{pmatrix} 4 & -2 & -2 \\ -2 & 5 & -1 \\ -2 & -2 & 4 \end{pmatrix}$$

$$\overline{L}_{P_1} = \begin{pmatrix} 6 & -2 & -2 \\ -2 & 6 & -2 \\ -2 & -2 & 8 \end{pmatrix} \quad \overline{L}_{P_2} = \begin{pmatrix} 7 & -2 & -1 \\ -2 & 4 & -2 \\ -1 & -2 & 7 \end{pmatrix} \quad \overline{L}_{P_3} = \begin{pmatrix} 6 & -2 & -2 \\ -2 & 5 & -1 \\ -2 & -2 & 6 \end{pmatrix}$$

Here we denote the matrix $(\cdot)_{P_j}$ to be the matrix in $\mathbb{R}^{11 \times 11}$ but with nonzero entries on $P_j$ only.

3. **Operator compression.** As mentioned in the section 1, inspired by the finite element method (FEM) approach for solving PDEs in which the variational formulation naturally gives the energy decomposition of the operator, we adopt a similar strategy of FEM to find a subspace $\Phi$ approximating the solution space of a linear system involving $A$ that are energy decomposable. In particular, approximation of $A^{-1}$ can also be obtained.

For traditional FEM, the accuracy of these approximations relies on the regularity of the given coefficients. Without assuming any smoothness on coefficients, one promising way to approximate the operator is to consider projecting the operator into the modified subspace $\Psi = \Phi - P_{\overline{U}}\Phi$ in [22] or $\Psi = A^{-1}(\Phi)$ as proposed in [27, 16].

Here $U = (\Phi)^{\perp}$ is the $l_2$-orthogonal complement space, and $P_{\overline{U}}$ is the A-orthogonal projection operator. In the case where $A$ is invertible, these two modified spaces are equivalent. Therefore, we propose to employ a similar methodology for compressing a general SPD matrix $A$. 

---

**Fig. 2.** (a) An illustration of a graph example. (b) An illustration of a partition $P = \{\{1, 2, 3\}, \{4, 5, 6, 7\}, \{8, 9, 10, 11\}\}$. 


We first obtain a general error estimate for projecting the matrix $A$ into a subspace $\Psi = A^{-1}(\Phi)$ of $\mathbb{R}^n$, given the projection-type approximation property of the subspace $\Phi$. With this observation, the operator compression problem is narrowed down into choosing an appropriate $\Phi$ which satisfies condition (11). The following lemma also gives us a general idea on how we can control the errors introduced during the compression of the operator $A$.

**Lemma 3.1.** Let $\Phi$ be a subspace of $\mathbb{R}^n$ and $P_\Phi$ be the orthogonal projection onto $\Phi$ with respect to $\langle \cdot, \cdot \rangle_2$. Let $\Psi$ be the subspace of $\mathbb{R}^n$ given by $\Psi = A^{-1}(\Phi)$ and $P_\Psi^A$ be the orthogonal projection onto $\Psi$ with respect to $\langle \cdot, \cdot \rangle_A$. If

$$\|x - P_\Phi x\|_2 \leq \epsilon \|x\|_A \quad \forall x \in \mathbb{R}^n$$

for some $\epsilon > 0$, then

1. for any $x \in \mathbb{R}^n$ and $b = Ax$, we have

   $$\|x - P_\Psi^A x\|_A \leq \epsilon \|b\|_2;$$

2. for any $x \in \mathbb{R}^n$ and $b = Ax$, we have

   $$\|x - P_\Psi^A x\|_2 \leq \epsilon^2 \|b\|_2;$$

3. and thus we have

   $$\|A^{-1} - P_\Psi^A A^{-1}\|_2 \leq \epsilon^2.$$

Interchangeably, we will use $\Phi$ and $\Psi$ to denote the basis matrix of the space $\Phi$ and $\Psi$, respectively, such that $\Phi^T \Phi = I_N$ and $\Psi = A^{-1} \Phi T$. Here $N$ is the dimension of $\Phi$, and $T$ is some $N \times N$ nonsingular matrix to be determined. Then we have

$$P_\Psi^A = \Psi (\Psi^T A \Psi)^{-1} \Psi^T A = A^{-1} \Phi (\Phi^T A^{-1} \Phi)^{-1} \Phi^T,$$

which is the $A$-orthogonal projection matrix into the subspace $\Psi$. In [22], Målqvist and Persson proposed the use of modified coarse space in order to handle roughness of coefficients when solving elliptic equations with FEM. Assuming that the finite elements are conforming and if we see $\Phi$ as the original coarse space $V_H$ in [22], then $\Psi$ is exactly the modified coarse space $V_{ms}^H$ as they proposed, and the first error estimate in Lemma 3.1 is consistent with their error analysis. More generally, Owhadi in [28] makes use of the gamblet framework to construct the basis of modified coarse space such that the conforming properties of those bases are no longer required. In particular, (15) is an analogy of $\Psi \Phi = K \Psi^T (\Phi K \Phi^T)^{-1} \Phi$ on page 9 of [27], and the error estimate in (12) is correspondingly the Proposition 3.6 in that paper.

As an FEM-type method, the choice of $\Phi$ determines the operator compression error or the solution approximation error. We know that the optimal rank-$N$ approximator of $A^{-1}$ is given by taking $\Phi$ to be the eigenspace of $A$ corresponding to the first $N$ smallest eigenvalues, which is essentially the principal component analysis (PCA) [17]. And the optimal compression error is given by $\epsilon^2 = (\lambda_{N+1}(A))^{-1}$. However, with the optimal approximation property, the drawback of the PCA is nonnegligible in that the eigenvectors of $A$ are almost always dense even when $A$ has strong local properties. While the sparse PCA [23, 42, 14] provides a strategy to obtain a sparse approximation of $A^{-1}$, it implicitly assumes that the operators inherit low-rank characteristics such that the $l_1$ minimization approach is effective. To fully use the local
properties of $A$, we would prefer to choose $\Phi$, which can be locally computed but still has a good approximation property, namely, satisfying condition (11) with a pretty good error $\epsilon$ and a nearly optimal dimension $N$. Also we hope the a priori error bound $\epsilon$ can be estimated locally.

Indeed when solving elliptic PDEs using FEM, the nodal basis can be chosen as (discretized) piecewise polynomials with compact local supports, and the error is given by the resolution of the partition of the computational domain [3, 8]. However, such choices of partition of the computational domain do not depend on the operator $A$ in traditional FEM. Yet it only depends on the geometry of the computational domain, and thus the performance relies on the regularity of $A$. So a natural question arises: Can we do better if we choose the partition and the nodal basis using the local information of $A$?

Furthermore, what can we do if we do not a priori have the computational domain? Such a scenario arises, for example, when the underlying geometry of some operators $A$, like graph Laplacian, is unknown and no embedding maps to the physical domain can be found easily. In this case, one of the promising ways to accomplish such a task lies in the deep connection between our energy representation of the operator $A$ and its hidden geometric structure. More specifically, the energy decomposition of the operator introduced in subsection 2.1 reveals the intrinsic locality of the underlying geometry in an algebraic way, so that we can construct an optimal partition of the computational space and choose a proper subspace/basis $\Phi$ using only local information.

After constructing the partition and the basis $\Phi$, the next mission is to find a good basis $\Psi$ of the space $A^{-1}(\Phi)$. The choice of $\Psi$ serves to preserve the locality of the stiffness matrix $A_{st} = \Psi^T A \Psi$ inherited from $A$ and to give a reasonable bound on the condition number of $A_{st}$.

In summary, as mentioned in section 1, our approach is to (i) construct a partition of the computational space/basis using local information of $A$, (ii) construct $\Phi$ that is locally computable in each patch of the partition and satisfies error condition (11), and (iii) construct $\Psi$ that provides stiffness matrix $A_{st}$ with locality and a reasonable condition number. The whole process can be summarized as the Algorithm 1, and each step will be discussed in following sections.

**Algorithm 1. Operator Compression.**

**Input:** Energy decomposition $E$, underlying basis $V$, desire accuracy $\epsilon$  
1: Construct partition $\mathcal{P}$ subject to $\epsilon$ using Algorithm 4;  
2: Construct $\Phi$ using Algorithm 2;  
3: Construct $\Psi$ using Algorithm 3 subject to $\epsilon$;  
4: Compute $P_{\Psi}A^{-1} = \Psi(\Psi^T A \Psi)^{-1} \Psi^T$ as the compressed operator.

But to theoretically develop our approach, we first assume that we are given an imaginary partition $\mathcal{P}$ and then derive proper constructions of $\Phi$ and $\Psi$ serving the desired purposes based on this partition. In the derivation process, we come up with some desired conditions that will, in return, guide us how to construct the adaptive partition $\mathcal{P}$ with the desirable properties.

### 3.1. Choice of $\Phi$.

As discussed in the last subsection, the underlying geometry of the operator may not be given. Therefore, determining $\Phi$, which archives the condition (11), is not a trivial task. Instead of tackling this problem directly, the following proposition provides us a more apparent and local criterion on choosing $\Phi$. 


Proposition 3.2. Let $\mathcal{P} = \{P_j\}_{j=1}^M$ be a partition of $\mathcal{V}$ and $\{A_{P_j}\}_{j=1}^M$ be the corresponding interior energies as defined in Definition 2.3. For each $1 \leq j \leq M$, let $\Phi_j$ be some subspace of $\text{span}\{P_j\}$ such that

$$\|x - P_{\Phi_j}x\|_2 \leq \epsilon \|x\|_{A_{P_j}} \quad \forall x \in \text{span}\{P_j\}$$

for some constant $\epsilon$. Then we have

$$\|x - P_{\Phi}x\|_2 \leq \epsilon \|x\|_A \quad \forall x \in \mathbb{R}^n,$$

where $\Phi = \bigoplus_j \Phi_j$.

Proof. Since $\mathcal{P} = \{P_j\}_{j=1}^M$ is a partition of $\mathcal{V}$, we have $P_{\Phi} = \sum_j P_{\Phi_j}$, and thus

$$\|x - P_{\Phi}x\|_2^2 = \left\| \sum_j (P_jx - P_{\Phi_j}x) \right\|_2^2 = \sum_j \|P_jx - P_{\Phi_j}x\|_2^2 \leq \epsilon^2 \sum_j \|P_jx\|_{A_{P_j}}^2.$$ 

Notice that

$$\sum_j \|P_jx\|_{A_{P_j}}^2 = \sum_j \|x\|_{A_{P_j}}^2 \leq \|x\|_A^2,$$

and the conclusion follows. \qed

Intuitively, given a partition $\mathcal{P}$ of $\mathcal{V}$, we can construct $\Phi$ locally by choosing $\Phi_j$ that satisfies (16) for each $P_j$. Apparently, the choice of $\Phi_j$ depends on the partition $\mathcal{P}$, and the feasibility of this problem is guaranteed since we can always set $\mathcal{P} = \mathcal{V}$ to fulfill (16). But this choice is not optimal. We should adaptively choose $\mathcal{P}$ and $\Phi$ in such a way that it minimizes $N$, the dimension of $\Phi$.

Suppose we are given the partition $\mathcal{P} = \{P_j\}_{j=1}^M$ (in other words, the number of patches, $M$, is fixed); minimizing $N$ is equivalent to minimizing the dimension of each $\Phi_j$. In the following, we will first define the notion of interior spectrum of interior energy $A_S$. Lemma 3.4 will then show the relationship between the interior spectrum and the minimum dimension that can be achieved for each $\Phi_j$.

Definition 3.3 (interior spectrum). Let $S$ be a subset of $\mathcal{V}$. We define the interior spectrum $\Lambda_{\text{int}}(S; A)$ as the set of eigenvalues of $A_S$, where $A_S$ is the interior energy of $S$ with respect to $A$ (or we can view it as an operator restricted to the space $\text{span}\{S\}$).

In what follows, since $A$ is generally given and fixed, we will write $\Lambda_{\text{int}}(S; A)$ as $\Lambda_{\text{int}}(S)$. Also, we will simply use $0 \leq \lambda_1(S) \leq \lambda_2(S) \leq \cdots \leq \lambda_s(S)$ to denote the ordered elements of $\Lambda_{\text{int}}(S)$, where $s = #S = \dim(\text{span}\{S\})$.

Lemma 3.4. Given an $S \subseteq \mathcal{V}$ and a constant $\epsilon$, let $q(\epsilon)$ be the smallest integer such that $\frac{1}{q(\epsilon)} \leq \lambda_{q(\epsilon)+1}(S)$. Also define $G(\epsilon) = \{\Theta \subseteq \text{span}\{S\} : \|x - P_{\Theta}x\|_2 \leq \epsilon \|x\|_A, \forall x \in \text{span}\{S\}\}$, and let $p(\epsilon) = \min_{\Theta \in G(\epsilon)} \dim \Theta$. Then we have $q(\epsilon) = p(\epsilon)$.

By Lemma 3.4, one optimal way to minimize $\dim \Phi_j$ for each $P_j$ subject to condition (16) is to take $\Phi_j = \Phi_j^{q(\epsilon)}$, the eigenspace corresponding to interior eigenvalues $\lambda_1(P_j) \leq \lambda_2(P_j) \leq \cdots \leq \lambda_{q(\epsilon)}(P_j)$, where $q(\epsilon)$ is the smallest integer such that $\frac{1}{q(\epsilon)} \leq \lambda_{q(\epsilon)+1}(P_j)$. Recall that this criterion for choosing $\Phi_j$ is based on the fact that the partition $\mathcal{P}$ is given. Then one shall ask a more practical question: How do we construct an “optimal” partition $\mathcal{P}$ in the sense that it has a smallest total dimension of $\Phi$?

Instead of answering this question directly, we consider the problem in a more tractable way. We fix an integer $q$ and choose a $q$-dimensional local space $\Phi_j$ for each
Algorithm 2. Construction of $\Phi$.

Input: Energy decomposition $\mathcal{E}$, partition $\mathcal{P}$ subject to $\varepsilon(\mathcal{P}, q) \leq \epsilon$.

1: for each $P_j \in \mathcal{P}$ do
2: Extract $A_{P_j}$ from $\mathcal{E}$;
3: Find the first $q$ normalized eigenvectors of $A_{P_j}$ as $\Phi_j$;
4: Find $U_j$ such that $[\Phi_j, U_j]$ is an orthonormal basis of $\text{span}(P_j)$;
5: end for
6: Collect all $\Phi_j$ as $\Phi$, and all $U_j$ as $U$.

Then the problem of minimizing $\dim \Phi$ subject to the condition (16) is reduced to finding a partition $\mathcal{P} = \{P_j\}_{j=1}^M$ with a minimal patch number. Still guided by Lemma 3.4, we know that we should choose $\Phi_j = \Phi_j^q$, and the condition (16) is satisfied if and only if $\frac{1}{\sqrt{\lambda_j+1}} \leq \lambda_{q+1}(P_j)$ for each $P_j$. Define the error factor of a partition $\mathcal{P}$ as

\begin{equation}
\varepsilon(P_j, q) = \frac{1}{\sqrt{\lambda_{q+1}(P_j)}}, \quad 1 \leq j \leq M, \quad \text{and} \quad \varepsilon(\mathcal{P}, q) = \max_j \frac{1}{\sqrt{\lambda_{q+1}(P_j)}},
\end{equation}

and so given a constant $\epsilon$, we need to minimize the patch number of $\mathcal{P}$ subject to $\varepsilon(\mathcal{P}, q) \leq \epsilon$.

**Construction 1 (construction of $\Phi$).** We choose $\Phi = \bigoplus_{j=1}^M \Phi_j^q$, where $\Phi_j^q \subset \text{span}(P_j)$ is the eigenspace corresponding to the first $q$ interior eigenvalues of patch $P_j$. We also require $(\Phi_j^q)^T \Phi_j^q = I_q$, i.e., $\Phi^T \Phi = I_N$. Then the condition (16) is satisfied if $\varepsilon(\mathcal{P}, q) \leq \epsilon$.

We propose Algorithm 2 to construct $\Phi$ guided by Construction 1. Notice that it also computes the compliment space $U_j$ of $\Phi_j$ in each span($P_j$), which will serve for the purpose of performing multiresolution matrix decomposition in section 6.

**Remark 3.5.**
- The construction of $U_j$ can be implicitly done, for example, by extending $\Phi_j$ to an orthonormal basis of span($P_j$) with local QR factorization, where only $q$ Householder vectors $[h_1, h_2, \ldots, h_q]$ need to be stored. In fact, we can apply economic QR factorization to $\Phi_j$ to obtain $$(I-h_1h_1^T)(I-h_2h_2^T)\cdots(I-h_qh_q^T) = [Q_j, U_j],$$ where $[Q_j, U_j]$ is orthogonal and $\text{Span}(\Phi_j) = \text{Span}(Q_j)$. In the following algorithms there are only two kinds of operation that involve $U_j$, namely, $U_j^T x$ for some $x \in \mathbb{R}^s$ and $U_j x$ for some $x \in \mathbb{R}^{s-q}$. The former one can be done by computing $y = (I-h_1h_1^T)(I-h_2h_2^T)\cdots(I-h_qh_q^T)x$ and then taking the last $s-q$ entries of $y$; the latter one can be done by extending $x$ to $\tilde{x} = [0, x]$ with additional $q$ 0’s in front and then computing $$(I-h_1h_1^T)(I-h_2h_2^T)\cdots(I-h_qh_q^T)\tilde{x}.$$
- The integer $q$ is given before the partition is constructed, and the choice of $q$ will be discussed in section 4.

**Complexity of Algorithm 2.** For simplicity, we assume that all patches in partition $\mathcal{P}$ have the same patch size $s$. Then number of patches is $\#\mathcal{P} = \frac{n}{s}$. Let $F(s)$ denote the local patchwise complexity of solving the partial eigenproblem and extending $\Phi_j$ to $[\Phi_j, U_j]$. Then the complexity of Algorithm 2 is

\begin{equation}
O\left(\frac{F(s)}{s} \cdot n\right).
\end{equation}
3.2. Choice of $\Psi$. Suppose that we have determined the space $\Phi = [\varphi_1, \varphi_2, \ldots, \varphi_N]$; the next step is to find $\Psi = [\psi_1, \psi_2, \ldots, \psi_N] = A^{-1} \Phi T$, namely, to determine $T$, so that

1. each $\psi_i$ is locally computable or can be approximated by some $\hat{\psi}_i$ that is locally computable;
2. the stiffness matrix $A_{st} = \Psi^T A \Psi$ has a relatively small condition number, or the condition number can be bounded by some local information.

Generally, each $A^{-1} \psi_i$ is not local (sparse), so it may be impossible to find even one $\psi \in \text{span}\{A^{-1} \Phi\}$ that is locally computable. A more promising idea is to find $\hat{\psi}$ that can be well approximated by some $\tilde{\psi}_i$ which is locally computable.

**Lemma 3.6.** Assume that $\Psi = [\psi_1, \psi_2, \ldots, \psi_N]$ satisfies $\|x - P^A_{\Psi} x\|_A \leq \epsilon \|Ax\|_2$ and $\|A_{st}^{-1}\|_2 \leq \|A^{-1}\|_2$ and that $\tilde{\Psi} = [\tilde{\psi}_1, \tilde{\psi}_2, \ldots, \tilde{\psi}_N]$ satisfies $\|\psi_i - \tilde{\psi}_i\|_A \leq \frac{C\epsilon}{\sqrt{N}}$, $1 \leq i \leq N$ for some constant $C$. Then

1. for any $x \in \mathbb{R}^n$ and $b = Ax$, we have
   $$\|x - P^A_{\Psi} x\|_A \leq (1 + C\|A^{-1}\|_2)\epsilon \|b\|_2;$$
2. for any $x \in \mathbb{R}^n$ and $b = Ax$, we have
   $$\|x - P^A_{\Psi} x\|_2 \leq (1 + C\|A^{-1}\|_2)^2 \epsilon^2 \|b\|_2;$$
3. and thus we have
   $$\|A^{-1} - P^A_{\Psi} A^{-1}\|_2 \leq (1 + C\|A^{-1}\|_2)^2 \epsilon^2;$$

Guided by Lemma 3.6, in order to preserve the compression accuracy, we require that each $\psi_i$ be approximated accurately in energy norm $\|\cdot\|_A$ by some $\tilde{\psi}_i$ that is locally computable. To implement this idea, we consider the problem reversely. Suppose we already have some $\tilde{\psi}_i$ that is locally computable, so the construction of $\Psi$ is to find $\hat{\psi}_i \in A^{-1}(\Phi)$ so that $\|\psi_i - \hat{\psi}_i\|_A$ is small for each $i$. Since $\tilde{\psi}_i$ is given, minimizing $\|\psi_i - \hat{\psi}_i\|_A$ can be simply solved by taking $\hat{\psi}_i = P^A_{\Psi} \tilde{\psi}_i$. Thanks to the expression $P^A_{\Psi} = A^{-1}(\Phi T A^{-1}\Phi)^{-1} \Phi T$, we can perform the energy projection $P^A_{\Psi}$ as long as we know $\Phi$. Therefore, we have

$$\begin{align*}
(20) \quad & \Psi = P^A_{\Psi} \tilde{\Psi} = A^{-1}\Phi (\Phi T A^{-1}\Phi)^{-1} \Phi T \tilde{\Psi} \\
(21) \quad & \Rightarrow \Phi T \Psi = \Phi T A^{-1}\Phi (\Phi T A^{-1}\Phi)^{-1} \Phi T \tilde{\Psi} = \Phi T \tilde{\Psi}.
\end{align*}$$

Then we shall discuss how to describe the locality of each $\tilde{\psi}_i$. Similar to the locality of $\Phi$, though it seems greedy, we can also require that $\psi_i \in \text{span}\{P_{j_i}\}$ for some $j_i$, and this requirement implies that $\varphi_{j_i}^T \psi_i = 0$ for all $\varphi_{j_i} \notin \text{span}\{P_{j_i}\}$. Then to determine $\Phi^T \Psi = \Phi T \tilde{\Psi}$, we still need to determine $\varphi^T_{j_i} \psi_i$ for each $\varphi_{j_i} \in \text{span}\{P_{j_i}\}$. But actually, in the following proof of exponential decay of $\psi_i$, we can see that the value of $\varphi^T_{j_i} \psi_i$ for each $\varphi_{j_i} \in \text{span}\{P_{j_i}\}$ does not essentially change the decay property of $\psi_i$. We only need to make sure that $\tilde{\Psi}$ has the same dimension as $\Phi$. So for simplicity, we require that

$$\varphi^T_{i'} \psi_i = \delta_{i', i}, \quad 1 \leq i' \leq N, \quad \text{i.e.,} \quad \Phi^T \Psi = \Phi^T \tilde{\Psi} = I_N.$$

Adding this extra localization constraint to the form of $\Psi = A^{-1} \Phi T$, we can choose $\Psi$ as follows.
CONSTRUCTION 2 (construction of Ψ). We choose Ψ = A^{−1}T so that Φ^T Ψ = I_N, that is,
\begin{equation}
(23) \quad Ψ = A^{−1}Φ(Φ^T A^{−1}Φ)^{−1}, \quad T = (Φ^T A^{−1}Φ)^{−1},
\end{equation}
and we have
\begin{equation}
(24) \quad A_{st} = Ψ^T A Ψ = (Φ^T A^{−1}Φ)^{−1}.
\end{equation}

Remark 3.7. Our choice of Ψ is inspired by the result proposed by Owhadi in [27], where the author obtained the same format of Ψ from a marvelous probabilistic perspective. In this work, the idea of *gamblet transformation* is introduced. Such transformation gives a particular choice of basis in the modified coarse space, which ensures the exponential decay feature of Ψ. Our derivation of the choice of Ψ can be seen as a algebraic interpretation of Owhadi’s probabilistic construction.

Though we construct each ψ_{i} from some local vector ˜ψ_i, the error \|ψ_t - ˜ψ_i\|_A is not necessarily small. To have both good locality and small error, we need to use something in between. The following lemma (see also section 3.2 in [27]) shows that the construction of Ψ in (23) is equivalent to the optimizer of a minimization problem.

**Lemma 3.8.** Let Ψ be constructed as in (23). Then for each i, ψ_{i} satisfies
\begin{equation}
ψ_{i} = \arg \min_{x \in \mathbb{R}^n} \|x\|_A,
\end{equation}
subject to \(ψ^T x = δ_{i,i'}\; \forall \; i' = 1, \ldots, N.

**Proof.** Notice that \(Aψ_{i} \in \Phi; thus, for any x that satisfies \(ψ^T x = δ_{i,i'}\), we have \(Φ^T (x - ψ_{i}) = 0\) and \(ψ_{i}^T A(x - ψ_{i}) = 0\). Then we have
\begin{equation}
(25) \quad ||x||_A = ||x - ψ_{i} + ψ_{i}||_A^2 = ||ψ_{i}||_A^2 + ||x - ψ_{i}||_A^2 + 2ψ_{i}^T A(x - ψ_{i}) ≥ ||ψ_{i}||_A^2.
\end{equation}

By the construction given in (23) and guided by Lemma 3.8, we can obtain every ψ_{i} by solving the optimization problem. Our next step is to make use of this minimal property to construct local ˜ψ_i that will be proved exponentially convergent to ψ_{i}.

**Definition 3.9** (layers of neighbors). Let \(P = \{P_j\}_{j=1}^J\) be a partition of \(\mathcal{V}\). For any \(P_j \in \mathcal{P}\), we recursively define \(S_0(P_j) = P_j\) and
\begin{equation}
(26) \quad S_{k+1}(P_j) = \bigcup_{P_j' \sim S_k(P_j)} P_j', \; k = 0, 1, 2, \ldots.
\end{equation}
\(S_k(P_j)\) is called the kth neighbor patch ball of \(P_j\) and \(S_k(P_j)/S_{k-1}(P_j)\) the kth neighbor patch layer of \(P_j\).

**Remark 3.10.** By making use of the notion of *neighboring* introduced in Definition 2.2, we can construct the “algebraic neighbor layers” starting from any initial patch \(P_j\). Still we do not implicitly assume any underlying physical domain to the operator \(A\).

**Definition 3.11** (local approximator). For each \(ψ_{i}\), let \(P_{j_i}\) be the patch such that \(ψ_{i} \in \Phi_{j_i} \subset \text{span}\{P_{j_i}\}\). Then for each \(k \geq 0\), we define the k-local approximator of \(ψ_{i}\) as
\begin{equation}
ψ^k_{i} = \arg \min_{x \in \text{span}\{S_k(P_{j_i})\}} \|x\|_A,
\end{equation}
subject to \(ψ^T x = δ_{i,i'}\; \forall \; i' = 1, \ldots, N.\)
Remark 3.12. Here $k$ is called the radius of $\psi_i^k$. The condition $\varphi_i^k = \delta_{i,k}$ is equivalent to $\Phi^T \psi_i^k = \Phi^T \psi_i$. By Lemma 3.8 and the definition of $\psi_i$, we have

\begin{align}
(\psi_i^k - \psi_i)^T A \psi_i = 0, \quad (\psi_i^{k-1} - \psi_i^k)^T A \psi_i = 0 \quad \forall k,
\end{align}

and hence

\begin{align}
\|\psi_i^k\|_A^2 = \|\psi_i\|_A^2 + \|\psi_i - \psi_i\|_A^2,
\end{align}

\begin{align}
\|\psi_i^{k-1}\|_A^2 = \|\psi_i^k\|_A^2 + \|\psi_i^k - \psi_i\|_A^2.
\end{align}

Definition 3.13 (condition factor of partition). Let $P = \{P_j\}_{j=1}^M$ be a partition of $V$. Writing $A_P = A_{P_j}^{-1}$, the inverse of $A_{P_j}$, as an operator restricted on span$\{P_j\}$, we define

\begin{align}
\delta(P, \Phi) = \max_{P_j \in \mathcal{P}} \frac{x^T x}{x^T A_{P_j} x}, \quad 1 \leq j \leq M, \quad \text{and} \quad \delta(P) = \max_{P_j \in \mathcal{P}} \delta(P_j, \Phi_j).
\end{align}

Remark 3.14.

- In what follows, since we always fix a choice of $\Phi$ for a partition $P$, we will simply use $\delta(P_j)$ and $\delta(P)$ to denote $\delta(P_j, \Phi_j)$ and $\delta(P, \Phi)$, respectively. In particular, when we use Construction 1 for $\Phi$ with some integer $q$, we correspondingly use the notations $\delta(P, q)$.
- If we follow the construction $\Phi_j = I_{q_j}$, where $q_j$ is the dimension of $\Phi_j$, then we have

\begin{align}
\delta(P_j, \Phi_j) = \max_{c \in \mathbb{R}^{q_j}} \frac{c^T \Phi_j^T \Phi_j c}{c^T \Phi_j^T A_{P_j}^{-1} \Phi_j c} = \max_{c \in \mathbb{R}^{q_j}} \frac{c^T c}{c^T \Phi_j^T A_{P_j}^{-1} \Phi_j c} = \|(\Phi_j^T A_{P_j}^{-1} \Phi_j)^{-1}\|_2,
\end{align}

that is,

\begin{align}
(\Phi_j^T A_{P_j}^{-1} \Phi_j)^{-1} \preceq \delta(P_j) I_{q_j} \preceq (\delta(P)) I_{q_j}.
\end{align}

Moreover, by blockwise inequalities, we have

\begin{align}
\left(\Phi^T \left(\sum_{j=1}^M A_{P_j}\right)^{-1}_1 \Phi\right)^{-1} \preceq (\delta(P)) I_N.
\end{align}

This analysis will help us to bound the maximum eigenvalue of the stiffness matrix $A_{st} = \Phi^T A \Phi$ by $\delta(P)$.

Example 3.15. In this example, we consider the operator $A$ to be the discretization of a two-dimensional (2-D) second-order elliptic operator by standard 5-point finite difference scheme. Similar to the case of graph Laplacian in Example 2.10, we have a natural energy decomposition inherited from the assembling of such discretization. Specifically, for every pair of vertices $e_{\text{hori}} := [(i,j), (i,j+1)]$ and $e_{\text{vert}} := [(i,j), (i+1,j)]$ in the finite difference grid, the energy elements are

\begin{align}
E_{\text{hori}}^{ij} = \frac{1}{|e_{\text{hori}}|^2} \begin{pmatrix}
0 & (i,j) \\
-a_{i,j+\frac{1}{2}} & -a_{i,j+\frac{1}{2}} \\
-a_{i,j+\frac{1}{2}} & \ddots & \ddots \\
& -a_{i,j+\frac{1}{2}} & a_{i,j+\frac{1}{2}} & 0
\end{pmatrix}.
\end{align}
Fig. 3. Example showing the relationship between mesh size, the error factor $\epsilon(P_j,1)$, the condition factor $\delta(P_j,1)$, and contrast.

Now suppose we are given a partition $\mathcal{P}$; we focus on a particular local patch $P_j$ to study how mesh size and contrast affect the error factor and the condition factor. Figure 3(a) shows the high-contrast field (colored in black) in $P_j$. For simplicity, we set $P_j$ as a square domain with fixed length $H = |e_{\text{hori}}| = |e_{\text{vert}}|$. We also set the coefficient in high-contrast field to be $10^3$ (and 1 otherwise). Figure 3(b) shows the decreasing trend of the error factor $\epsilon(P_j,1)$ as the vertex number $#V$ inside the patches (i.e., the vertex density in $P_j$) increases. Here we choose $q = 1$ for illustration purposes. Figure 3(c) shows a similar decreasing trend of the condition factor $\delta(P_j,1)$, and Figure 3(d) plots $\epsilon(P_j,1)^2 \cdot \delta(P_j,1)$ versus $#V$. Fixing the vertex number $#V$ in the patch $P_j$, we also study the relationship of $\epsilon(P_j,1)$, $\delta(P_j,1)$ and the contrast. In particular, we double the contrast by 2 in each single computation and investigate the trend of $\epsilon(P_j,1)$ and $\delta(P_j,1)$. Figure 3(e) shows the decrease of $\epsilon(P_j,1)$ as contrast increases. For $\delta(P_j,1)$, although it also increases as the contrast increases, we can clearly see that there is an upper bound (around 220 in this example), even when the contrast jumps up to $2^{20}$. Figure 3(g) plots $\epsilon(P_j,1)^2 \cdot \delta(P_j,1)$ versus contrast.

The following theorem shows the scaling properties of $\psi_i, \psi_k^i$ under construction Construction 2 and Definition 3.11, which will help to prove the exponential decay of the basis function $\psi_i$. 

and

$$E_{vert}^{ij} = -\frac{1}{|e_{vert}|^2} \begin{pmatrix} 0 & (i+1,j) \\ a_{i+\frac{1}{2},j} & -a_{i+\frac{1}{2},j} \\ -a_{i+\frac{1}{2},j} & a_{i+\frac{1}{2},j} \\ 0 & (i,j) \end{pmatrix}.$$
THEOREM 3.16. For each $\psi_i$, we have

\[
\|\psi_i\|_A \leq \|\bar{\psi}_i^k\|_A \leq \|\psi_i^0\|_A \leq \sqrt{\delta(P_j)}.
\]

Compliment space. For each $P_j$, without causing any ambiguity, we use $U_j$ interchangeably to denote both the orthogonal compliment of $\Phi_j$ with respect to $\text{span}\{P_j\}$ and the orthonormal basis matrix of $U_j$. Namely, we have $U_j \subset \text{span}\{P_j\}$ and $\Phi_j^T U_j = 0$. Then we define

\[
\alpha(P_j) = \max_{x \in U_j} \frac{x^T \bar{A}_{P_j} x}{x^T A_{P_j} x}, \quad 1 \leq j \leq M, \quad \text{and} \quad \alpha(P) = \max_{P_j \in P} \alpha(P_j).
\]

Remark 3.17.

- If we choose $\Phi_j$ so that it satisfies condition (16), then we have

\[
x^T \bar{A}_{P_j} x = \|x\|^2_{\bar{A}_{P_j}} \geq \frac{1}{c^2} \|x - P_{\Phi_j} x\|^2 = \frac{1}{c^2} \|x\|^2 \quad \forall x \in U_j
\]

and \(x^T \bar{A}_{P_j} x \leq \|A_{P_j}\| \|x\|^2\). Thus, \(\alpha(P_j) \leq c^2 \|A_{P_j}\|\). This argument is meant to show that we can have \(\alpha(P) < +\infty\) if we choose $P$ and $\Phi$ properly. But this bound is not tight, as \(\alpha(P)\) can be much smaller in general.

- An immediate result of the definition of $\alpha(P_j)$ is that

\[
U_j^T \bar{A}_{P_j} U_j \leq \alpha(P_j) U_j^T \bar{A}_{P_j} U_j \quad \forall j.
\]

The following theorem shows that the local basis function $\bar{\psi}_i^k$ is exponentially convergent to $\psi_i$ as its support $S_k(P_j)$ extends (or as $k$ increases). Indeed, the exponential decay of $\psi_i$ has been proved in [22, 27, 28, 16] in different manners based on a common observation that the energy of $\psi_i$ in the region beyond a certain single layer of patches is comparable to its energy only on this layer, which reflects the local interacting feature of the operator $A$ itself. Also based on this observation, we modify the proof in section 6 of [28] using a matrix framework coherent to our energy settings.

THEOREM 3.18 (exponential decay). For each $\psi_i$, we have

\[
\|\psi_i^k - \psi_i\|^2_A \leq \left( \frac{\alpha(P) - 1}{\alpha(P)} \right)^k \|\psi_i^0 - \psi_i\|^2_A \leq \left( \frac{\alpha(P) - 1}{\alpha(P)} \right)^k \delta(P_j).
\]

Proof. For simplicity, we will write $\psi_i$ as $\psi_i^k$, $P_j$ as $P$, and $S_k(P_j)$ as $S_k$. Let $Y_k$ denote the joint space of all $U_j$ such that $P_j \subset S_k$ and $Z_k$ the joint space of all $U_j$ such that $P_j \subset V \setminus S_k$. We still use $Y_k, Z_k$ as the basis matrix for the spaces $Y_k, Z_k$, so that each $U_j$ is a bunch of columns of either $Y_k$ or $Z_k$. We use $U$ to denote $Y_\infty$. Notice that we can always arrange $U_j$ in a particular order so that the matrix form $U = [Y_k, Z_k]$ holds. We define

\[
r^k = \psi^k - \psi, \quad k \geq 0; \quad w^k = \psi^{k-1} - \psi, \quad k \geq 1;
\]

then according to (30), we have

\[
\|r^{k-1}\|^2_A = \|r^k\|^2_A + \|w^k\|^2_A.
\]
Since $\Phi^T(\psi^k - \psi) = \Phi^T(\psi^{k-1} - \psi^k) = 0$, we have $r^k \in U$ and $w^k \in Y_k$. Then by the minimal properties of $\psi^k$ and $\psi$, we actually have

$$r^{k-1} = P^A_U \psi^{k-1} = U(U^T A U)^{-1} U^T A \psi^{k-1}$$

(40)

$$w^k = P^A_U \psi^{k-1} = Y_k(Y_k^T A Y_k)^{-1} Y_k^T A \psi^{k-1} = Y_k(Y_k^T A S_k Y_k)^{-1} Y_k^T A S_k \psi^{k-1}.$$  

By the definition of $S_k$, we know that $S_{k-1} \neq V \setminus S_k$, and therefore $Z_k^T A \psi^{k-1} = 0$. Then we get

$$\|r^{k-1}\|_A = \psi^{k-1, T} A U (U^T A U)^{-1} U^T A \psi^{k-1}$$

$$= \psi^{k-1, T} A \left[ Y_k \ 0 \right] (U^T A U)^{-1} \left[ Y_k^T \ 0 \right] A \psi^{k-1}.$$ 

Due to the locality of $\overline{\mathcal{A}}_{P_j}$, we obtain

$$U^T A U \succeq U^T \left( \sum_{j=1}^{M} \overline{\mathcal{A}}_{P_j} \right) U = \sum_{j=1}^{M} \left( 0 \ U_j^T A_{P_j} U_j \ 0 \right) \succeq \frac{1}{\alpha(\mathcal{P})} \sum_{j=1}^{M} \left( 0 \ U_j^T A_{P_j} U_j \ 0 \right).$$

As a simple inference of Proposition 2.9, we have

$$\sum_{P_j \subset S_k} \overline{\mathcal{A}}_{P_j} \succeq A_{S_k}, \quad \sum_{P_j \subset V \setminus S_k} \overline{\mathcal{A}}_{P_j} \succeq A_{V \setminus S_k},$$

and therefore

$$\sum_{j=1}^{M} \left( 0 \ U_j^T A_{P_j} U_j \ 0 \right) \succeq \left( Y_k^T A_{S_k} Y_k \ z_k^T (\sum_{P_j \subset V \setminus S_k} \overline{\mathcal{A}}_{P_j}) z_k \right).$$

Combining all results, we have

$$(U^T A U)^{-1} \preceq \alpha(\mathcal{P}) \left( \begin{array}{c} Y_k^T A_{S_k} Y_k \\ Z_k^T A_{V \setminus S_k} Z_k \end{array} \right)^{-1}$$

$$= \alpha(\mathcal{P}) \left( \begin{array}{c} (Y_k^T A_{S_k} Y_k)^{-1} \\ (Z_k^T A_{V \setminus S_k} Z_k)^{-1} \end{array} \right).$$

and thus

$$\|r^{k-1}\|_A^2 \leq \alpha(\mathcal{P}) \psi^{k-1, T} A \left[ Y_k \ 0 \right] \left( Y_k^T A_{S_k} Y_k \right)^{-1} \left( Z_k^T A_{V \setminus S_k} Z_k \right)^{-1} \left[ Y_k^T \ 0 \right] A \psi^{k-1}$$

$$= \alpha(\mathcal{P}) \psi^{k-1, T} A Y_k (Y_k^T A_{S_k} Y_k)^{-1} Y_k^T A \psi^{k-1}$$

$$= \alpha(\mathcal{P}) \psi^{k-1, T} A S_k Y_k (Y_k^T A_{S_k} Y_k)^{-1} Y_k^T A_{S_k} \psi^{k-1}$$

$$= \alpha(\mathcal{P}) \|w^k\|_A^2.$$ 

This gives us

$$\|r^{k-1}\|_A^2 \leq \|r^k\|_A^2 + \|w^k\|_A^2 \leq \|r^k\|_A^2 + \frac{1}{\alpha(\mathcal{P})} \|r^{k-1}\|_A^2 \implies \|r^k\|_A^2 \leq \frac{\alpha(\mathcal{P}) - 1}{\alpha(\mathcal{P})} \|r^{k-1}\|_A^2.$$
Applying this recursively, we have

\[
\|\psi_k - \psi\|_A^2 \leq \left( \frac{\alpha(P) - 1}{\alpha(P)} \right)^k \|\psi^0 - \psi\|_A^2.
\]

Notice that \(\|\psi^0 - \psi\|_A^2 = \|\psi_0\|_A^2 - \|\psi\|_A^2 \leq \|\psi^0\|_A^2 \leq \delta(P)\), and this completes our proof.

Remark 3.19.

- Recall that in Lemma 3.6, to make a \(k\)-layer approximator \(\Psi^k\) become a good approximator, we need \(\|\psi_i - \psi_i^k\|_A \leq \frac{C\epsilon}{\sqrt{N}}\) for some constant \(C\), and thus Theorem 3.18 guides us to choose

\[
k = O \left( \log \frac{1}{\epsilon} + \log N + \log \delta(P) \right).
\]

And the locality of \(\psi_i^k\) lies in the local connection property of the matrix \(A\).

- By the definition in (36), \(\alpha(P)\) is locally scaling invariant. Therefore, the layerwise decay rate is unchanged when \(A\) is locally multiplied by some scaling constant.

Corollary 3.20. For any \(\psi_i\), the interior energy of \(\psi_i\) on \(V/S_k(P_{ji}) = S_k^c(P_{ji})\) decays exponentially with \(k\), namely,

\[
\|\psi_i\|_{A(S_k^c(P_{ji}))}^2 \leq \left( \frac{\alpha(P) - 1}{\alpha(P)} \right)^k \delta(P_{ji}).
\]

Moreover, for any two \(\psi_i, \psi_{i'}\), we have

\[
|\psi_i^T A \psi_{i'}| \leq \left( \frac{\alpha(P) - 1}{\alpha(P)} \right)^{\frac{k_{i'i'} - 1}{2}} \delta(P),
\]

where \(k_{i'i'}\) is the largest integer such that \(P_{ji'} \subset S_{k_{i'i'}}(P_{ji})\) (or equivalently \(P_{ji} \subset S_{k_{i'i'}}^e(P_{ji'})\)).

Remark 3.21. Recall that in Lemma 3.6 for the compression error with localization \(\epsilon_{\text{com}}^2\) to be bounded by some prescribed accuracy \(\epsilon^2\), we need the localization error \(\epsilon_{\text{loc}}^2 \leq \frac{\epsilon^2}{N}\). But now with the exponential decaying feature of \(\Psi\), empirically, we observe that we can relax the requirement of the localization error to be \(\epsilon_{\text{loc}}^2 \leq O(\epsilon^2)\) in practice.

Now we have constructed a \(\Psi\) that can be approximated by a local computable basis in the energy norm. So the remaining task is to tackle with the second criterion: to give a control on the condition number of \(A_{st} = \Psi^T A \Psi\).

Theorem 3.22. Let \(\lambda_{\text{min}}(A_{st})\) and \(\lambda_{\text{max}}(A_{st})\) denote the smallest and largest eigenvalues of \(A_{st}\), respectively; then we have

\[
\lambda_{\text{min}}(A_{st}) \geq \lambda_{\text{min}}(A), \quad \lambda_{\text{max}}(A_{st}) \leq \delta(P),
\]

and so we have

\[
\kappa(A_{st}) = \frac{\lambda_{\text{max}}(A_{st})}{\lambda_{\text{min}}(A_{st})} \leq \delta(P)\|A^{-1}\|_2.
\]
COROLLARY 3.23. Let $\Psi$ be the local approximator of $\Psi$ defined in Lemma 3.6 such that $\|\psi_i - \tilde{\psi}_i\|_A \leq \frac{\epsilon}{\sqrt{N}}$; then we have

$$
\lambda_{\min}(\tilde{A}_{st}) \geq \lambda_{\min}(A), \quad \lambda_{\max}(\tilde{A}_{st}) \leq \left(1 + \frac{\epsilon}{\sqrt{\delta(P)}}\right)^2 \delta(P),
$$

and thus

$$
\kappa(\tilde{A}_{st}) = \frac{\lambda_{\max}(\tilde{A}_{st})}{\lambda_{\min}(\tilde{A}_{st})} \leq \left(1 + \frac{\epsilon}{\sqrt{\delta(P)}}\right)^2 \delta(P)\|A^{-1}\|_2.
$$

Guided by Theorem 3.22, we obtain a simple methodology on the control of condition number of the stiffness matrix $A_{st}$. As shown in (44) and Theorem 3.22, the only variable is the choice of partition $P$, and thus the burden again falls to the construction of the partition $P$. Nevertheless, this new criterion allows us to regulate the quality of partitions directly by avoiding large $\delta(P)$.

Now we can design an algorithm to construct the local approximator $\tilde{\Psi}$ of $\Psi$ subject to a desired localization error $\epsilon_{loc}$. Intuitively, a straightforward way is to choose a large enough uniform decay radius $r$ and directly compute $\tilde{\Psi} = \Psi^r$. The localization error can then be guaranteed by Lemma 3.6. But redundant computation will probably occur since some $\psi$ may decay much faster than the others. Instead, we propose to compute each $\tilde{\psi}_i$ hierarchically from the center patch $P_i$ by making use of optimization property (27). Suppose that we already obtain $\tilde{\psi}_i^{k-1}$; then by optimization property (27), one can check that $w_i^k = \tilde{\psi}_i^{k-1} - \tilde{\psi}_i^k$ satisfies the following optimization problem:

$$
\begin{align*}
& \text{arg min} \hspace{1em} w \in \text{span}\{S_i(P_j)\} \|\tilde{\psi}_i^{k-1} - w\|_A \\
& \text{subject to} \hspace{1em} \Phi^T w = 0.
\end{align*}
$$

(48)

Similar to the proof of Theorem 3.18, let $Y_{i,k}$ denote the joint space of all $U_j$ such that $P_j \subset \{S_i(P_j)\}$. Then the constraints in optimization problem (48) imply that $w_i^k \in \text{span}\{Y_{i,k}\}$. Therefore, we can explicitly compute $w_i^k$ as

$$
w_i^k = P_{Y_{i,k}}^A \tilde{\psi}_i^{k-1} = Y_{i,k} \left(Y_{i,k}^T A S_i(P_j) Y_{i,k}\right)^{-1} Y_{i,k}^T A S_i(P_j) \tilde{\psi}_i^{k-1},
$$

and thus

$$
\psi_i^k = \tilde{\psi}_i^{k-1} - Y_{i,k} \left(Y_{i,k}^T A S_i(P_j) Y_{i,k}\right)^{-1} Y_{i,k}^T A S_i(P_j) \tilde{\psi}_i^{k-1}.
$$

(50)

Specially, we can compute $\psi_i^0$ by (50) with an initial guess $\psi_i^{0-1} \in \text{span}\{P_j\}$ satisfying $\varphi_i^0 = \delta_{i',i} \forall \ i' = 1, \ldots, N$. Notice that the main cost of computation of $\psi_i^k$ involves inverting the matrix $Y_{i,k}^T A Y_{i,k}$, whose condition number can be bounded by $\epsilon(P,q)^2 \lambda_{\max}(A) \kappa(Y_{i,k}^T A Y_{i,k})$ as we will see in Lemma 6.1. By choosing all $U_j$ orthonormal, we have $\kappa(Y_{i,k}^T A Y_{i,k}) = 1$; then the computation efficiency is measured by $\epsilon(P,q)^2 \lambda_{\max}(A)$ if we use the CG-type method. When we prescribe some certain accuracy $\epsilon(P,q)^2$, but $\lambda_{\max}(A)$ is really large, so a multiresolution strategy will be adopted to ensure the efficiency of computing $\psi_i^k$.

To summarize, the process of computing a sufficient approximator $\tilde{\psi}_i$ starts with the formation of $\psi_i^0$, then inductively computes $\psi_i^k$ by solving inverse problem (50)
with initializer \( \psi_i^{k-1} \), and finally ends with \( \tilde{\psi}_i = \psi_i^j \) when some stopping criterion is attained for \( k = r \). Such inductive computation suggests that we use the CG method to take advantage of the exponential convergence of \( \psi_i^k \). Having faith in the exponential decay of \( \| \psi_i^k - \psi_i \|_A \), we choose the stopping criterion as

\[
\eta_k^2 \frac{\| \psi_i^{k-1} - \psi_i^k \|_A^2}{1 - \eta_k^2} \leq \epsilon_{\text{loc}}^2 \text{ for } \eta_k = \frac{\| \psi_i^{k-1} - \psi_i^k \|_A}{\| \psi_i^{k-2} - \psi_i^{k-1} \|_A}.
\]

The reason is that if \( \| \psi_i^k - \psi_i \|_A \) does decay as \( \| \psi_i^k - \psi_i \|_A = O(\eta^k) \) for some constant \( \eta \in (0, 1) \), then

\[
\| \psi_i^k - \psi_i \|_A^2 = \frac{\eta^2}{1 - \eta^2} O(\| \psi_i^{k-1} - \psi_i \|_A^2 - \| \psi_i^k - \psi_i \|_A^2) = \frac{\eta^2}{1 - \eta^2} O(\| \psi_i^{k-1} - \psi_i \|_A^2),
\]

where we have used (30). With the analysis above, we propose Algorithm 3 for constructing \( \Psi \).

**Algorithm 3. Construction of \( \tilde{\Psi} \).**

**Input:** Energy decomposition \( \mathcal{E} \), partition \( \mathcal{P} \), \( \Phi \), desired accuracy \( \epsilon_{\text{loc}} \)

**Output:** \( \tilde{\Psi} \)

1. for \( i = 1, 2, \ldots, \dim(\Phi) \) do
2. Compute \( \psi_i^0 \) by solving Equation (50) on \( S_0(P_j) \);
3. Compute \( \tilde{\psi}_i \) by solving Equation (50) on \( S_1(P_j) \) with initializer \( \psi_i^0 \);
4. repeat
5. Compute \( \psi_i^k \) by solving Equation (50) on \( S_k(P_j) \) with initializer \( \psi_i^{k-1} \);
6. \( \eta \leftarrow \frac{\| \psi_i^k - \psi_i^{k-1} \|_A}{\| \psi_i^{k-1} - \psi_i \|_A} \);
7. until \( \frac{\eta^2}{1 - \eta^2} \| \psi_i^k - \psi_i \|_A^2 < \epsilon_{\text{loc}}^2 \);
8. \( \tilde{\psi}_i \leftarrow \psi_i^k \);
9. end for
10. \( \tilde{\Psi} = [\tilde{\psi}_0, \tilde{\psi}_1, \ldots, \tilde{\psi}_{\dim(\Phi)}] \).

**Complexity of Algorithm 3.** For simplicity, we assume that all patches in partition \( \mathcal{P} \) have the same patch size \( s \). Let \( r \) be the necessary number of layers for \( \| \psi_i^r - \psi_i \|_A \leq \frac{\epsilon}{\sqrt{N}} \), where \( N \) is the dimension of \( \Phi \) (or \( \Psi \)); then we have

\[
r = O \left( \log \frac{1}{\epsilon} + \log N + \log \delta(\mathcal{P}) \right).
\]

Since we are actually compressing \( A^{-1} \), we can bound \( N \) by the original dimension \( n \). Further, we assume that locality conditions (56), (57), and (58) are satisfied; then the support size of each \( \psi_i^r \) is \( O(s \cdot r^d) \). Since we also only need to solve (50) up to the same relative accuracy \( O(\frac{\epsilon}{\sqrt{N}}) \) using the CG method, the cost of computing \( \psi_i^r \) can be estimated by

\[
O \left( k(Y_i^{r,k} A Y_i^{r,k}) \cdot s \cdot r^d \cdot \left( \log \frac{1}{\epsilon} + \log N + \log \delta(\mathcal{P}) \right) \right) \leq O \left( \varepsilon(\mathcal{P}, q)^2 \cdot \lambda_{\max}(A) \cdot s \cdot \left( \log \frac{1}{\epsilon} + \log n + \log \delta(\mathcal{P}) \right)^{d+1} \right).
\]
Finally, the total complexity of Algorithm 3 is $N$ times the cost for every $\psi_\nu^r$, i.e.,

$$O \left( n \cdot q \cdot \varepsilon(P, q)^2 \cdot \lambda_{\text{max}}(A) \cdot \left( \log \frac{1}{\varepsilon} + \log n + \log \delta(P) \right)^{d+1} \right),$$

where we have used the relation $N = nq/s$.

**4. Construction of partition $P$.** With the analysis in the previous sections, we now have a blueprint for the construction of partition $P = \{P_j\}_{j=1}^M$. Given an underlying energy decomposition $E = \{E_k\}_{k=1}^m$ of an SPD matrix $A$ and an orthonormal basis $V$ of $\mathbb{R}^n$, the basic idea is to find a partition $P$ of $V$ with small patch number $\#P$ and small condition factor $\delta(P, q)$ while subject to a prescribed error bound on the error factor $\varepsilon(P, q)$. In particular, our goal is to find the optimizer of the following problem:

$$P = \arg \min_{\tilde{P}} f_1(\#\tilde{P}) + f_2(\delta(\tilde{P}, q))$$

subject to $\varepsilon(\tilde{P}, q) \leq \varepsilon$,

where $f_1, f_2$ are some penalty functions, $q$ is a chosen integer, and $\varepsilon$ is the desired accuracy. This ideal optimization problem is intractable since in general such discrete optimization means to search over all possible combinations. Instead, we propose to use the local clustering approach to ensure efficiency.

Generally, if we have a priori knowledge of the underlying computational domain of the problem, like $\Omega \subset \mathbb{R}^d$, one of the optimal choices of partition will be the uniform regular partition. For instance, in [8], regular partitions are used in the sense that each patch (finite element) has a circumcircle of radius $H$ and an inscribed circle of radius $\rho H$ for some $\rho \in (0, 1)$. The performance under regular partitioning relies on the regularity of the coefficients of $A$ (low contrast, strong ellipticity), and the equivalence between the energy norm defined by $A$ and some universal norm independent of $A$. In particular, since regular partitioning of the computational domain is simply constructed regardless of the properties of $A$, its performance cannot be ensured when $A$ loses some regularity in some local or microscaled regions.

In view of this, a more reasonable approach is to construct a partition $P$ based on the information extracted from $A$, which is represented by the local energy decomposition $E$ of $A$ in our proposed framework. For computational efficiency, the construction procedure should rely only on local information (rather than global spectral information as in the procedure of eigendecomposition). This explains why we introduce the local measurements in section 3: the error factor $\varepsilon(P, q)$ and the condition factor $\delta(P, q)$, which keep track of the performance of partition in our searching approach. These measurements are locally (patchwisely) computable and thus provide the operability of constructing partition with local operations interacting with only neighbor data.

To make use of the local spectral information, we propose to construct the desired partition $P$ of $V$ by iteratively clustering basis functions in $V$ into patches. In particular, small patches (sets of basis) are combined into larger ones, and the scale of the partition becomes relatively coarser and coarser. For every such newly generated patch $P_j$, we check if $\varepsilon(P_j, q)$ still satisfies the required accuracy (see (16)). The whole clustering process stops when no patch combination occurs, that is, when the partition achieves the resolution limit. Also, for patch $P_j$ to be well-conditioned, we set a bound $c$ on $\delta(P_j, q) \varepsilon(P_j, q)^2$. The motivation of such a bound will be explained in section 6. And for large $\delta(P_j, q)$ to diminish, patches with a large condition factor...
Algorithm 4. Pair-Clustering.

**Input:** energy decomposition $E$, underlying basis $V$, desired accuracy $\epsilon$, condition bound $c$.

**Output:** Partition $P$.

1. **Initialize:** $P_j = \{v_j\}$, $\delta(P_j, q) = A_{P_j}$ (scalar), $1 \leq j \leq n$;
2. **while** Number of active patches $> 0$, **do**
3. **Sort** active $\{P_j\}$ with respect to $\delta(P_j, q)$ in descending order;
4. Mark all (active and inactive) $P_j$ as unoperated;
5. **for** each active $P_j$ in descending order of $\delta(P_j, q)$, **do**
6. Find $\text{Match}(P_j, \epsilon, c)$;
7. **if** $\text{Find\_Match}$ succeeds, **then**
8. Mark $P_j$ as operated;
9. **else if** all neighbor patches of $P_j$ are unoperated, **then**
10. Mark $P_j$ as inactive;
11. **end if**
12. **end for**
13. **end while**

Algorithm 5. $\text{Find\_Match}$.

**Input:** $P_j$, $\epsilon$, $c$.

**Output:** Succeeds or Fails.

1. **for** $P_{j''} \sim P_j$ **do**
2. Find largest $\text{Con}(P_j, P_{j''})$ among all unoperated $P_{j''}$ (stored as $P_{j''}$);
3. **end for**
4. Compute $\varepsilon(P_j \cup P_{j''}, q)$ and $\delta(P_j \cup P_{j''}, q)$;
5. **if** $\varepsilon(P_j \cup P_{j''}, q) \leq \epsilon$ & $\delta(P_j \cup P_{j''}, q) \varepsilon(P_j \cup P_{j''}, q)^2 \leq c$, **then**
6. combine $P_j$ and $P_{j''}$ to form $P_j$ ($P_{j''}$ no longer exists);
7. update $\delta(P_j, q)$;
8. **return** $\text{Find\_Match}$ succeeds.
9. **else**
10. **return** $\text{Find\_Match}$ Fails.
11. **end if**

are combined first. To realize the partitioning procedure and maintain the computation efficiency, we combine patches pairwisely. Our proposed clustering algorithm is summarized in Algorithms 4 and 5.

Remark 4.1.

- If we see $1/\varepsilon(P_j \cup P_{j''}, q)^2$ as the gain and $\delta(P_j \cup P_{j''}, q)$ as the cost, the well-conditioning bound $\varepsilon(P_j \cup P_{j''}, q)^2 \delta(P_j \cup P_{j''}, q) \leq c$ implies that the cost is proportional to the gain.
- If we want the patch sizes to grow homogeneously, we can take patch size into consideration when sorting the patches (line 3 in Algorithm 4).
- The local basis functions, $\Phi_j$, are also computed in the subfunction $\text{Find\_Match}$ and can be stored for future use.

The subfunction $\text{Find\_Match}$ in line 6 of Algorithm 4 takes a patch $P_j$ as input and finds another patch $P_{j''}$ that will be absorbed by $P_j$. As a local operation, the possible candidates for $P_{j''}$ are just the neighboring patches of $P_j$. To further accelerate
the algorithm, we avoid checking the error factor for all possible pair \((P_j, P_{j'})\) with \(P_j \sim P_{j'}\). Alternatively, we check the patch \(P_{j'}\) that has the largest “connection” (correlation) with \(P_j\). Undoubtedly, this quantity can be defined in different ways. Here we propose the connection between \(P_j\) and \(P_{j'}\) as

\[
\text{Con}(P_j, P_{j'}) = \sum_{E \sim P_j, E \sim P_{j'}} \left( \sum_{u \in P_j, v \in P_{j'}} |u^T E v| \right).
\]

On the one hand, note that \(\text{Con}(P_j, P_{j'})\) can be easily computed and inherited directly after patch combination since one can check that \(\text{Con}(P_j \cup P_{j'}, P_{j''}) = \text{Con}(P_j, P_{j''}) + \text{Con}(P_{j'}, P_{j''})\). On the other hand, we observe that

\[
\Delta_{P_j \cup P_{j'}} = \Delta_{P_j} + \Delta_{P_{j'}} + \sum_{E \sim P_j \cup P_{j'}} E = \Delta_{P_j} + \Delta_{P_{j'}} + \text{Cross Energy}.
\]

In other words, a larger cross energy implies larger interior eigenvalues of \(\Delta_{P_j \cup P_{j'}}\), which means \(\Delta_{P_j \cup P_{j'}}\) is less likely to violate the accuracy requirement. One can also recall the similarity of this observation to the findings in spectral graph theory, where stronger connectivity of the graph corresponds to larger eigenvalues of the graph Laplacian \(L\). These motivate us to simplify the procedure by examining the patch candidate \(P_{j'}\) with the largest connection to \(P_j\).

Though our algorithm does not assume any a priori structural information of \(A\), its efficiency and effectiveness may rely on the hidden locality properties of \(A\). To perform a complexity analysis of Algorithm 4, we first introduce some notations. Similar to the layers of neighbors defined in Definition 3.9, we define \(N_1(v) = \mathcal{N}(v)\) and

\[
N_{k+1}(v) = \mathcal{N}(N_k(v)) = \{u \in \mathcal{V} : u \sim N_k(v)\};
\]

that is, for any \(u \in N_k(v)\), there is a path of length \(k\) that connects \(u\) and \(v\) with respect to the connection relation “\(\sim\)” defined in Definition 2.2. The following definition describes the local interaction property of \(A\).

**Definition 4.2** (locality/sparsity of \(A\)). A is said to be local of dimension \(d\) with respect to \(\mathcal{V}\) if

\[
\#N_k(v) = O(k^d) \quad \forall k \geq 1, \quad \forall v \in \mathcal{V}.
\]

The following definition describes the local spectral properties of an energy decomposition of \(A\). It states that a smaller local patch corresponds to a smaller scale and that \(\varepsilon(P, q)\) tends to increase and \(\delta(P, q)\) tends to decrease as the patch size of \(P\) increases. This explains why we combine patches from finer scales to coarser scales to construct the desired partition \(\mathcal{P}\).

**Definition 4.3** (local energy decomposition). \(\mathcal{E} = \{E_k\}_{k=1}^m\) is said to be a local energy decomposition of \(A\) of order \((q, p)\) with respect to \(\mathcal{V}\) if there exists some constant \(h\) such that

\[
\varepsilon(N_k(v), q) = O((hk)^p) \quad \forall k \geq 1, \quad \forall v \in \mathcal{V}.
\]

Moreover, \(\mathcal{E}\) is said to be well-conditioned if there is some constant \(c\) such that

\[
\varepsilon(N_k(v), q)^2 \delta(N_k(v), q) \leq c \quad \forall k \geq 1, \quad \forall v \in \mathcal{V}.
\]
Remark 4.4.

- The locality of $A$ implies that $\#\nabla_{k+1}(v) - \#\nabla_k(v) = O(d \cdot k^{d-1})$. In particular, $\#\nabla_1(v) = O(d)$, and thus the number of nonzero entries of $A$ is $m = O(d \cdot n)$.

- Let $P$ be a partition of $\mathcal{V}$ such that each patch $P \in \mathcal{P}$ satisfies $\text{diam}(P) = O(r)$ and $\#\mathcal{P} = O(r^d)$, where $r$ is an integer and “diam” is the path diameter with respect to the adjacency relation “~” defined in Definition 2.2. Let $S_k(P)$ be the layers of neighbors (patch layers) defined in Definition 3.9 and $\#^P S_k(P)$ denote the number of patches in $S_k(P)$. Then the locality of $A$ implies that

$$\#^P S_k(P) = O\left(\frac{\# S_k(P)}{r^d}\right) = O\left(\frac{(rk)^d}{r^d}\right) = O(k^d).$$

This means that a $\mathcal{V}$ with adjacency relation defined by $A$ has a self-similar property between fine scale and coarse scale.

These abstract formulations/notations actually summarize a large class of problems of interest. For instance, suppose $A$ is assembled from the FEM discretization of a well-posed elliptic equation with homogeneous Dirichlet boundary conditions:

$$Lu = \sum_{0 \leq |\sigma|, |\gamma| \leq p} (-1)^{|\sigma|} D^\sigma (a_{\sigma\gamma} D^\gamma u) = f, \quad u \in H_0^p(\Omega),$$

where $\Omega \subset \mathbb{R}^d$ is a bounded domain. Let $\mathcal{V}$ be the nodal basis of the discretization and each energy element $E$ in $\mathcal{E}$ be the energy inner product matrix (i.e., the stiffness matrix) of the neighbor nodal functions on a fine mesh patch. The locality of $L$ and the underlying dimension of $\Omega$ ensure that $A$ is local of dimension $d$ with respect to $\mathcal{V}$. With a consistent discretization of $L$ on local domains, the interior energy corresponds to a Neumann boundary condition, while the closed energy corresponds to a Dirichlet boundary condition. In this sense, using a continuous limit argument and the strong ellipticity assumption, Hou and Zhang in [16] prove that if $q \geq (p+d-1)/d$, then $\varepsilon(N_k(v), q) \lesssim (hk)^p$ (generalized Poincaré inequality), where $h$ is the fine mesh scale and $p$ is half the order of the elliptic equation and $\varepsilon(N_k(v), q)^2 \delta(N_k(v), q) \lesssim c$ (inverse estimate) for some scaling-invariant constant $c$. Unfortunately, these arguments would be compromised if strong ellipticity were not assumed, especially when high-contrast coefficients are present. However, as we see in Example 3.15, $\varepsilon(N_k(v), q)$ and $\delta(N_k(v), q)$ actually converge when the contrast of the coefficient becomes large, which is not explained by general analysis. So we could still hope that the matrix $A$ and the energy decomposition $\mathcal{E}$ have the desired locality that can be numerically learned, even when conventional analysis fails.

**Estimate of patch number.** Intuitively, if $A$ is local of dimension $d$ and $\mathcal{E}$ is well-conditioned and local of order $q$, then the patch number of an ideal partition $\mathcal{P}$ subject to accuracy $\epsilon$ should be

$$\#\mathcal{P} = O\left(\frac{n}{(\epsilon^{1/p}/h)^d}\right) = O\left(\frac{nh^d}{\epsilon^{d/p}}\right), \quad \delta(\mathcal{P}, q) = O\left(\frac{c}{\epsilon^2}\right),$$

where we estimate the path diameter of each patch in $\mathcal{P}$ by $O(\epsilon^{1/p}/h)$ and thus the patch size by $O((\epsilon^{1/p}/h)^d)$.

**Inherited locality.** As we have made the locality assumption on $A$, we would hope that the compressed operator $P_\psi A^{-1} = \tilde{\psi} \tilde{A}^{-1} \tilde{\psi}^T$ can also take advantage of
such locality. In fact, the localization of \( \Psi \) not only ensures the efficiency of the construction of \( \bar{\Psi} \) but also conveys the locality of \( A \) to the stiffness matrix \( \bar{A}_{st} \). Suppose \( A \) is local of dimension \( d \). Let \( \bar{\Psi} \) be the local approximator obtained in Algorithm 3 such that \( \bar{\psi}_i = \psi_i^r, 1 \leq i \leq N \) for some uniform radius \( r \). Let \( \bar{V} = \{ \bar{v}_i \}_{i=1}^N \) be the orthonormal basis of \( \mathbb{R}^N \) such that

\[
\bar{v}_i^T \bar{A}_{st} \bar{v}_j = \bar{v}_i^T \bar{\Psi}^T A \bar{\Psi} \bar{v}_j = \psi_i^r T \bar{\Psi}^T A \bar{\Psi} \bar{v}_j \quad \forall 1 \leq i, j \leq N.
\]

We then similarly define the adjacency relation between basis vectors in \( \bar{V} \) with respect to \( \bar{A}_{st} \). Since each localized basis \( \bar{\psi}_i \) interacts with patches in \( r \) patch layers (thus interacts with other \( \bar{\psi}_j \) corresponding to patches in \( 2r \) layers) and each patch corresponds to \( q \) localized basis functions, using the result in Remark 4.4, we have

\[
\#N_k(\bar{v}) = O((rk)^d \cdot q) = q \cdot r^d \cdot O(k^d) \quad \forall k \geq 1 \quad \forall \bar{v} \in \bar{V}.
\]

That means \( \bar{A}_{st} \) inherits the locality of dimension \( d \) from \( A \). In addition, by using the same argument as in Remark 4.4, we have \( \#N_k(\bar{v}) = O(d \cdot q \cdot r^d) \), which is the number nonzero (NNZ) entries of one single column of \( \bar{A}_{st} \). Therefore, the NNZ entries of \( A_{st} \) can be bounded by \( O(N \cdot d \cdot q \cdot r^d) = O(m \cdot q^2 \cdot r^d/s) \), where \( m = O(d \cdot n) \) is the NNZ of \( A \), \( s \) is the average patch size, and we have used the relation \( N = nq/s \). In particular, if the localization error is subject to \( \epsilon_{loc} = \frac{1}{\sqrt{N}} \), then the radius has estimate \( r = O((\log \frac{1}{\epsilon} + \log n + \log \delta(P,q))) \), and thus the bound on the NNZ of \( \bar{A}_{st} \) becomes

\[
O \left( m \cdot q^2 \cdot \frac{1}{s} \cdot \left( \log \frac{1}{\epsilon} + \log n + \log \delta(P,q) \right)^d \right).
\]

**Choice of \( q \).** Recall that, instead of choosing a larger enough \( q_j \) for each patch \( P_j \) to satisfied \( \epsilon(P_j, q_j) \leq \epsilon \), we use a uniform integer \( q \) for all patches and leave the mission of accuracy to the construction of partition. So before we proceed to the algorithm, we still need to know what \( q \) we should choose. In some problems, \( q \) can be determined by theoretical analysis. For example, when solving elliptic equation of order \( p \) with FEM, we should at least choose \( q = \left( \frac{p+1}{2d} - 1 \right) \) to obtain an optimal rate of convergence. And as for graph Laplacians that are generally considered as a discrete second-order elliptic problem \( (p = 1) \), we can thus choose \( q = 1 \). But when the problem is more complicated and has no intrinsic order, the choice of \( q \) can be tricky. So one practical strategy is to start from \( q = 1 \) and increase \( q \) when the partition obtained is not acceptable.

**Complexity of Algorithm 4.** For simplicity, we assume that all patches in the final output partition \( P \) have the same patch size \( s \). Under locality assumption of \( A \) given in (56), the local operation cost of **Find_Match**\((P_j)\) is approximately

\[
O(d \cdot F(size(P_j))) \leq O(d \cdot F(s)).
\]

Here \( F(\#P_j) \) is a function of \( \#P_j \) that depends only on complexity of solving local eigenproblems (with respect to \( \bar{A}_{P_j} \) ) and local inverse problems (with respect to \( \bar{A}_{P_j} \ ) on patch \( P_j \); thus, we can bound \( F(s) \) by \( O(s^3) \). Since patches are combined pairwisely the number of while-loops starting at line 2 is of order \( O(\log s) \), and in each while-loop, the operation cost can be bounded by

\[
O \left( d \cdot \frac{F(s)}{s} \cdot n \right) + O(n \cdot \log n) = O(d \cdot s^2 \cdot n) + O(n \cdot \log n),
\]
where \( O(d \cdot \frac{F(s)}{s} \cdot n) \) comes from operating Find_Match\((P_j)\) for all surviving active patch \( P_j \) and \( O(n \cdot \log n) \) comes from sorting operations. Therefore, the total complexity of Algorithm 4 is

\[
O \left( d \cdot s^2 \cdot \log s \cdot n \right) + O(\log s \cdot n \cdot \log n).
\]

**Complexity of Algorithm 1.** Combining all procedures together and noticing that Algorithm 2 can be absorbed to Algorithm 4, the complexity of Algorithm 1 is

\[
O \left( d \cdot s^2 \cdot \log s \cdot n \right) + O(\log s \cdot n \cdot \log n)
\]

\[+ O \left( q \cdot n \cdot \epsilon^2 \cdot \|A\|_2 \cdot \left( \log \frac{1}{\epsilon} + \log n + \log \delta(P) \right)^{d+1} \right),
\]

where \( n \) is the original dimension of basis, \( s \) is the maximal patch size, \( \epsilon \) is the prescribed accuracy, and we have used the fact \( \epsilon(P, q) \leq \epsilon \).

**Remark 4.5.**
- The maximum patch size \( s \) can be viewed as the compression rate since \( s \sim \frac{n}{M} \), where \( M \) is the patch number. The complexity analysis above implies that for a fixed compression rate \( s \), complexity of Algorithm 4 is linear in \( n \). However, when the locality conditions (56) and (57) are assumed, one can see that \( s \sim (\epsilon^{d}/h)^{d} \), where \( \epsilon \) is the desired (input) accuracy. As a consequence, while the desired accuracy is fixed, if \( n \) increases, then the complexity is no longer linear in \( n \) since \( h \) (finest scale of the problem) may change as \( n \) changes. In other words, Algorithm 4 loses the near linear complexity when the desired accuracy \( \epsilon \) is set too large compared to the finest scale of the problem. To overcome such limitation, we should consider a hierarchical partitioning introduced in section 6.
- As we mentioned before, the factor \( \epsilon^2 \|A\|_2 \) also suggests a hierarchical compression strategy, when \( \|A\|_2 \) is too large compared to the prescribed accuracy \( \epsilon^2 \).

5. **Numerical examples.** In this section, two numerical examples are reported to demonstrate the efficacy and effectiveness of our proposed operator compression algorithm. For consistency, all the experiments are performed on a single machine equipped with Intel Core i5-4460 CPU with 3.2 GHz and 8 GB DDR3 1600 MHz RAM.

5.1. **Numerical example 1.** The first numerical example arises from solving the finite graph Laplacian system \( Lx = b \), where \( L \) is the Laplacian matrix of a \( d \)-dimensional undirected random graph \( G = [V, E] \). Vertices \( x_i = (x_i^1, \ldots, x_i^d) \in V \subset \mathbb{R}^d \) are generated subject to a uniform distribution over the domain \( \Omega = [0, 1]^d \). Edge weights are then given by

\[
w_{ii} = c_i \quad \forall i; \quad w_{ij} = \begin{cases} \frac{r_{ij}^2}{\eta}, & \text{if } r_{ij}^2 \leq \eta/n^{\frac{d}{2}} \\ 0, & \text{otherwise} \end{cases} \quad \forall i \neq j,
\]

where \( r_{ij} = \|x_i - x_j\|_2 \), \( n = \#V \) is the number of vertices and \( \eta > 0 \) is some density factor for truncating long-distance interactions. We set \( c_i = 1 \) \( \forall i \) for the sake of well-posedness and invertibility of the graph Laplacian, which gives \( \|L^{-1}\|_2 = 1 \). We also remark that our choice of \( \eta \) ensures that the graph is locally connected and that the second smallest eigenvalue of \( L \) is of order \( O(1) \). This also gives \( n \propto \text{NNZ entries} \).
of $L$ in this example (which is actually not required by our algorithm). The basis $\mathcal{V} \subset \mathbb{R}^n$ is given by the natural basis with respect to vertices values, and the energy decomposition $\mathcal{E} = \{E_k\}_{k=1}^m$ is collected as described in Example 2.10, where each $E_k$ corresponds to an edge in $G$. Since $p = 1$ for graph Laplacian, we set $q = 1$ throughout this numerical example.

We first verify the complexity of Algorithm 4 by applying it to partition random graphs generated as described above. To be consistent, we set the prescribed accuracy $\frac{1}{d} \propto n$ and the upper bound $c$ of $\delta(\mathcal{P}, 1)\varepsilon(\mathcal{P}, 1)^2$ to be 100 in all cases, which is large enough for patches to combine with each other. Figure 4 illustrates the nearly linear time complexity of our algorithm with respect to the graphs’ vertex number $n$, which is consistent with our complexity estimation in subsection 4. Every dot represents the partitioning result of the given 2-D/three-dimensional (3-D) graph. In particular, the red and blue sets of dots are the partition results for 2-D graphs under the construction of $\eta = 2.5$ and $\eta = 3$, respectively, while the magenta point set represents the partitioning of the random 3-D graphs under the setting of $\eta = 4$. Similarly, Figures 4(b) and 4(d) shows, respectively, the time complexity of Algorithm 4 versus the NNZ entries of $L$. Notice that for graphs having NNZ entries with order up to $10^5$, the running time is still within seconds. These demonstrate the lightweight nature of Algorithm 4.

Second, we record the patch numbers of partition $\mathcal{P}$ obtained from Algorithm 4. We fix the domain $\Omega = [0, 1] \times [0, 1]$ and gradually increase the density of vertices in the graph. Therefore, we have $n \propto \frac{1}{\varepsilon}$, and thus $\#\mathcal{P} \propto \frac{1}{\varepsilon}$ by the observation in (59). The relationship between $\frac{1}{\varepsilon}$ and patch number $\#\mathcal{P}$ for the three cases is plotted in Figure 5(a) and 5(c), respectively. Figure 5(a) and 5(b) shows the linear relationship between $\#\mathcal{P}$ and $\frac{1}{\varepsilon^2}$, meaning that $d = 2$ in these cases. Similarly, the relationship

![Fig. 4. Illustration of the running time of Algorithm 4. (a) plots the running time against the number of vertices of random generated graphs in different dimensions and density factor $\eta$. (b)–(d) plot the running time against the number of NNZ entries in the graph Laplacian operators.](image)

![Fig. 5. Intrinsic dimension of the sets of graphs.](image)
plot in Figure 5(c) that discloses the dimension of the input graphs is 3-D as #\( P \propto \epsilon^3 \). These results precisely justify the capability of our framework in capturing geometric information of the domain.

Third, we focus on one particular 2-D random graph with vertex number \( n = 10000 \) to verify the performance of our algorithm on controlling the error and well-posedness of the corresponding compressed operator. We employ the concept of the \( k \)-nearest neighbor (KNN) to impose local interaction. Specifically, for each vertex \( x_i \), we denote \( \text{NN}(x_i, k_i) \) to be the set of the \( k_i \) nearest vertices of \( x_i \). Any two vertices \( x_i \) and \( x_j \) have an edge of weight \( w_{ij} = 1/r_{ij}^2 \) if and only if \( x_i \in \text{NN}(x_j; k_j) \) or \( x_j \in \text{NN}(x_i; k_i) \). Let \( y = (0.5, 0.5) \) be the center of \( \Omega \). We set \( k_i = 15 \) if \( \|x_i - y\|_2 \leq 0.25 \) and \( k_i = 5 \) otherwise. Therefore, the subgraph inside the disk \( B(y, 0.25) \) has a stronger connectivity than the subgraph outside.

We perform Algorithm 1 with a fixed condition control \( c = 50 \), and a prescribed accuracy \( \epsilon^2 \) varies from 0.001 to 0.0001. Figure 6(a) shows the ratios \( c_{\text{com}}^2/\epsilon^2 \) and \( \varepsilon(P, 1)^2/\epsilon^2 \), where \( c_{\text{com}}^2 = \|L^{-1} - P^T \Psi L^{-1}\|_2 \) is the compression error. Using Algorithm 4, we achieve a nearly optimal local error control. Also notice that the global compression error ratio \( c_{\text{com}}^2/\epsilon^2 \) is strictly bounded by 1 but also above 0.5, meaning that our approach is neither playing down nor overdoing the compression. Figure 6(b) shows the condition number of \( A_{st} \), which is consistent with the prescribed accuracy and is strictly bounded by \( \epsilon^2 \cdot \delta(P, 1) \) and the prescribed condition bound \( c \). Figure 6(c) plots patch number \#\( P \) versus \( \epsilon^{2}\) (the blue curve). Though the graph has different connectivity at different parts, it is still a 2-D graph and is locally connected in the sense of (56). Therefore, the curve is below linear, which is consistent with estimate (59) with \( d = 2 \). As a comparison, the red curve is the optimal compression dimension subject to the same prescribed accuracy given by eigenvalue decomposition. Since Algorithm 4 combines patches pairwisely, the output patch number \#\( P \) can be up to 2 times the optimal case. Figure 6(d) shows the partition result with \#\( P = 298 \) for the case \( \epsilon^2 = 0.001 \), where the black lines outline the boundaries of patches. Figure 6(e)
illustrates the patch sizes of the partition. We can see that patches near the center of the domain have larger sizes than the ones near the boundary since the graph has a higher connectivity inside the disk $B((0.5, 0.5), \frac{1}{4})$.

We also fix the prescribed accuracy $\varepsilon^2 = 0.0001$ to study the performance of compression with localization. In this case we have $N = \#P = \dim(\Phi) = 1446$. Let $\tilde{\Psi}$ be the local approximator of $\Psi$ constructed by Algorithm 3 subject to localization error $\|\tilde{\psi}_i - \psi_i\|^2_\Delta \leq \varepsilon^2_{\text{loc}}$. Figure 7 shows the compression error $\varepsilon^2_{\text{com}} = \|L^{-1} - P_\tilde{\Psi}L^{-1}\|_2$, the mean radius and the mean support size of $\tilde{\Psi}$ versus $\varepsilon^2_{\text{loc}}$ ranging from 0.1 to 0.0001. Recall that Lemma 3.6 requires a localization error $\varepsilon^2_{\text{loc}} = \varepsilon^2 / N$ to ensure $\varepsilon^2_{\text{com}} \leq \varepsilon^2$, but Figure 7(a) shows that $\varepsilon^2_{\text{loc}} = \varepsilon^2$ is adequate. Figure 7(b) shows the linearity between mean radius and $\log \frac{1}{\varepsilon^2_{\text{loc}}}$, which is consistent to the exponential decay of $\Psi$ proved in Theorem 3.18. Figure 7(c) shows the quadratic relation between mean support size and mean radius of $\tilde{\Psi}$, which again reflects that the geometric dimension of the graph is 2.

By fixing $\varepsilon^2_{\text{loc}} = 0.0001$, we have the mean radius of $\tilde{\Psi} \approx 4.5$ and the mean support size $\approx 449$. We pick three functions $\psi_1, \psi_2, \psi_3$ such that $\psi_1$ is close to the center of $\Omega$, $\psi_2$ is near the boundary of connectivity change, and $\psi_3$ is close to the boundary of $\Omega$. Figure 8(a) and 8(f) (first two rows of Figure 8) shows the profiles of $|\psi_i|$ and $\log_{10} |\psi_i|$ for $i = 1, 2, 3$. Though all of them decay exponentially from their center patches to outer layers, $\psi_1$ decays slower than $\psi_3$ since the graph has a higher connectivity (i.e., larger patch sizes) near the center. Figure 8(g) and 8(h) shows the profiles of $|\psi_i|$ and $\log_{10} |\psi_i|$ for $i = 1, 2, 3$. The bird's-eye view of their supports is shown in Figure 8(i). Similarly, $\psi_1$ needs a larger support than $\psi_3$ to achieve the same accuracy, which implies that $\psi_1$ decays slower than $\psi_3$. Figure 9 shows the spectrum of $L^{-1}$, $L^{-1} - P_\Psi L^{-1}$, and $L^{-1} - P_{\tilde{\Psi}} L^{-1}$. Notice that if we truncate the fine-scale part of $L^{-1}$ with prescribed accuracy $\varepsilon^2$, then $L^{-1} - P_{\tilde{\Psi}} L^{-1}$ has rank $n - N$, and the compression error is $8.13 \times 10^{-5} < \varepsilon^2$. Similarly, if the local approximator $\tilde{\psi}$ is applied (instead of $\psi$), then $L^{-1} - P_{\tilde{\Psi}} L^{-1}$ also has rank $n - N$, and the compression error is also $8.13 \times 10^{-5} < \varepsilon^2$. This means that the same compression error is achieved by the compression $P_{\tilde{\Psi}} L^{-1}$ with localization.

Fourth, we compare our results with the compression given by the PCA [17], that is, $L^{-1} \approx \sum_{i=1}^{N_{\text{PCA}}} \lambda_i^{-1} q_i q_i^T$, where $\lambda_i$ is the $i$th smallest eigenvalue of $L$ and $q_i$ is the corresponding normalized eigenvector. On the one hand, to achieve the same compression error $8.13 \times 10^{-5}$, we need $N_{\text{PCA}} = 893$, which is the optimal compression dimension for such accuracy. But we remark that such achievement requires solving a global eigenproblem, and the compressed operator $\sum_{i=1}^{N_{\text{PCA}}} \lambda_i^{-1} q_i q_i^T$ usually loses the original sparsity features. On the other hand, our approach has...
5.2. Numerical example 2. Our second numerical example arises from using GFEM to solve the following elliptic equation with homogeneous Dirichlet boundary conditions:

$$-\nabla \cdot (a \cdot \nabla u) = f, \quad u \in H^1_0(\Omega),$$

where $a$ is a coefficient function and $f \in L^2(\Omega)$. We use this equation to demonstrate the effectiveness of our approach in recovering small eigenvalues and eigenvectors of the Laplace operator $L$.

In the numerical example, we observe that the eigenvectors of $L$ corresponding to small eigenvalues can be recovered by solving the eigenvalue problem of the approximated system matrix $A_{st}$ or $\tilde{A}_{st}$. The eigenvectors are used to approximate the solution to the elliptic equation, and the procedure provides convenience for computing the first few eigenvalues and eigenvectors of $L$ since the approximated system has a much smaller condition number compared to the original system.

The process involves solving the eigenvalue problem of the approximated system matrix, which results in a smaller condition number and a faster computation time.

The diagrams illustrate the profiles of the eigenvectors and approximations, showing how the local functions are combined to form the approximations. The approximations are shown to be accurate, especially for small eigenvalues, as indicated by the close agreement with the exact eigenvectors.
Fig. 9. Spectrum of \( L^{-1} \), \( L^{-1} - P \Psi L^{-1} \), and \( L^{-1} - P \tilde{\Psi} L^{-1} \).

Fig. 10. Difference between the true eigenvalues/eigenvectors and the approximated ones.

Fig. 11. Plot of 2nd, 10th, 20th, and 50th eigenvectors corresponding to \( L^{-1} \) (first row) and \( \tilde{A}^{-1} \) (second row), respectively.
where $\Omega = [0, 1] \times [0, 1]$, $f \in L^2(\Omega)$, and the coefficient $a$ is a 2-by-2 matrix function of $(x, y)$ of the form

\begin{equation}
\begin{pmatrix}
a_{11} & a_{12} \\
a_{12} & a_{22}
\end{pmatrix} = \begin{pmatrix}
\cos \theta & \sin \theta \\
-\sin \theta & \cos \theta
\end{pmatrix} \begin{pmatrix}
\mu \cdot e_1 & 0 \\
0 & \mu \cdot e_2
\end{pmatrix} \begin{pmatrix}
\cos \theta & -\sin \theta \\
\sin \theta & \cos \theta
\end{pmatrix}.
\end{equation}

Here $\theta = \theta(x, y) \in C(\Omega)$ is the rotation (deformation) factor, $\mu = \mu(x, y) \in L^\infty(\Omega)$ is the contrast factor, and $e_i = e_i(x, y) \in L^\infty(\Omega)$, $i = 1, 2$ are the roughness factor. For the problem to be elliptic and well-posed, we require that $\mu e_1, \mu e_2 > C$ for some uniform constant $C > 0$. To increase the level of difficulty in solving this elliptic PDE, we choose $e_1$ and $e_2$ to be highly oscillatory, and $\mu$ varies from $O(1)$ to $O(10^6)$ (high contrasts). More precisely, $e_1, e_2$ are generated with extreme roughness as $e_i(x, y) = 1 + w_i(x, y)$, $i = 1, 2$, where for each point $(x, y)$, we set $w_1(x, y), w_2(x, y) \sim U([-0.1, 0.1])$, a uniform distribution on $[-0.1, 0.1]$. The contrast factor $\mu(x, y)$ is generated from the background permeability as shown in Figure 12(a). $\theta$ is given by $\theta(x, y) = \pi \cdot (x + y)$. The magnitude of $|a_{11}(x, y)|$ in $\Omega$ is also plotted in Figure 12(d) as a reference.

We use GFEM [8] with a regular triangularization to form a finite system that is fine enough to capture the details of the background field. The basis $V \in \mathbb{R}^n$ is the vector representation of the Galerkin nodal basis, and the SPD matrix $A$ is the stiffness matrix of nodal basis with respect to the energy inner product $\int_\Omega (\nabla \cdot \cdot)^T a(\nabla \cdot \cdot) dx dy$. 

![Figure 12](image-url)

**Fig. 12.** Partitioning result: operator stemmed from FEM of elliptic PDE with high-contrasted coefficients. (a) shows the distribution of the high-contrast factor in the domain $\Omega$. (b) and (c) show the regular partition in resolution $200 \times 200$ and $400 \times 400$, respectively. (d) show the value of $a_{11}(x, y)$ on $\Omega$. (e) and (f) show the partition results obtained from Algorithm 4.
The energy decomposition $\mathcal{E} = \{E_k\}_{k=1}^m$ is the collection of all patchwise stiffness matrices $E_k$ on every triangle $\tau_k$. Specifically, each $E_k$ has the form

$$E_k = \begin{pmatrix} 0 & w_{1k,2k} & w_{1k,3k} \\ w_{2k,1k} & w_{2k,2k} & w_{2k,3k} \\ w_{3k,1k} & w_{3k,2k} & w_{3k,3k} & 0 \end{pmatrix}, \quad w_{1k,jk} = \int_{\tau_k} (\nabla \phi_{i_1})^T a(\nabla \phi_{j_k}) dx dy, \quad i,j=1,2,3,$$

where $\phi_{i_k}$, $i=1,2,3$ are the three nodal basis surrounding $\tau_k$. In this case, every finest energy element involves three functions, which generalizes the concept of graphs’ edges as we mentioned before. One should also notice that for patch $\tau_k$ touching the boundary of $\Omega$, the corresponding $E_k$ reduces to involve only two functions or one function since nodal basis functions on the boundary are not required for homogeneous Dirichlet boundary conditions. Moreover, we will choose $q=1$ since the problem (63) is of second order (i.e., $p=1$).

In this example, we compare our partitioning technique with the performance of regular partition to illustrate the adaptivity of our algorithm to the features of the coefficient function $a(x,y)$. Here we consider the cases with two different resolutions, which are the regular triangulations with $200 \times 200$ and $400 \times 400$ vertices, respectively. We set $q=1$, the prescribed accuracy $\varepsilon^2 = 10^{-4}$, and the upper bound $c = 300$. We apply Algorithm 1 to compute the compressed operator for both cases.

In the case of regular partition, we use a uniform and regular partition on $\Omega$ that the coefficient function $a$ has the form

$$a(x,y) = \begin{cases} 1 & \text{for } d \leq 0.5 \\ 0 & \text{for } d > 0.5 \end{cases}$$

In this case, every eigenvector, $\Phi_j$, on every patch is a constant. Such choice of $\Phi$, along with the use of regular partition, is equivalent to the setup in [27, 16]. Therefore, the only modification we apply in this numerical example is the adaptive construction of the construction $P$, which remarkably improves the behavior of the compressed operator $A^{-1}$. We would also like to remark that in the case without high-contrasted channels, Algorithm 4 coherently gives a regular partitioning on the domain as conventional partition methods.

Table 1 summarizes the partitioning results in both cases. Under regular partitioning, we have $\#P = 1156$, and each $P_j$ has the size of at most $6 \times 6$ (Figure 12(b)) and $12 \times 12$ (Figure 12(c)) vertices, respectively. Notably, the condition factors for both cases go up to $10^{31}$, and the corresponding true condition numbers $\kappa(A_{\text{d}})$ are having an order of $10^8$, which show that such partition will produce an ill-posed compressed operator. Using our approach, the square error factors $\varepsilon(P,1)^2$ achieved in both cases are strictly bounded above by the prescribed accuracy $\varepsilon^2 = 10^{-4}$, and $\delta(P,1)$ is of the order of $10^6$ only. Indeed, the true compression error is even smaller, as the square error factor $\varepsilon(P,1)^2$ is only the theoretical upper bound as required in Proposition 3.2. Furthermore, the true condition numbers $\kappa(A_{\text{d}})$ are in the order

| Partition | $\#P$ | $\varepsilon(P,1)^2$ | $\delta(P,1)$ | $\kappa(A_{\text{d}})$ | $\delta(P,1)||A^{-1}||_2$ | max $\delta(P_j,1)^2$ | $\varepsilon(P_j,1)^2$ |
|-----------|-------|----------------------|----------------|-----------------|-----------------------------|---------------------|-------------------|
| 200 × 200 | Ours  | 11396 | $9.9932 \times 10^{-5}$ | $3.8703 \times 10^6$ | $4.8023 \times 10^7$ | $1.1462 \times 10^4$ | 233.3080 |
| Regular   | 1156  | $5.7733 \times 10^{-5}$ | $2.5299 \times 10^{11}$ | $2.5325 \times 10^{10}$ | $7.4923 \times 10^9$ | 454.4966 |
| 400 × 400 | Ours  | 1199  | $9.9958 \times 10^{-5}$ | $4.9467 \times 10^6$ | $3.1195 \times 10^7$ | $1.1073 \times 10^4$ | 277.6022 |
| Regular   | 1156  | $5.0450 \times 10^{-5}$ | $3.1410 \times 10^{13}$ | $2.2573 \times 10^9$ | $8.5947 \times 10^8$ | 1.0560 × 10^9 |
of $10^4$ (compared to $10^8$ from regular partitioning), which are again bounded by (and are much smaller than) $\delta(P, 1)\|A^{-1}\|_2$ as observed in Theorem 3.22. Moreover, the patch number $\#P$ is comparable to the case of regular partitioning. Also notice that $\max_{P_j \in \mathcal{P}} \delta(P_j, 1) \cdot \epsilon(P_j, 1)^2 < c = 300$ in both cases, which are coherent to the prescribed requirement. These results successfully illustrate the consistency between the numerical results and theoretical discoveries in the previous sections. The partition results obtained by Algorithm 4 are shown in Figure 12(e) and 12(f), respectively.

We remark that the reason for the huge difference in the condition number is caused by the nonadaptivity of the partitioning (i.e., regular partition) to the given operator $A$. Specifically, if some patches are fully covered in the high channel regions, the accuracy achieved by this patch is obviously very promising. However, the corresponding patchwise condition factor will jump up to the similar order as the high-contrast factor. In other words, patches which are fully covered by regions of high contrast should be avoid. As shown in Figure 12(f) and 12(e), our proposed Algorithm 4 can automatically extract the intrinsic geometric information of the operator (which is the distribution of high-contrast regions) and prevent patches which are fully enclosed in the high-contrast regions.

We also consider the profiles, log-profiles, and supports of three localized functions $\tilde{\psi}_1, \tilde{\psi}_2, \tilde{\psi}_3$ obtained by Algorithm 3 with prescribed localization error $\epsilon_{loc}^2 = 10^{-4}$ in the $200 \times 200$ resolution case. The plots are shown in Figure 13. We can see in Figure 13(a) that $\tilde{\psi}_1$ has multiple peaks (three peaks exactly), with one high-contrast channel cutting through. This means that this single function characterizes the local feature of the operator. Though the exponential decaying feature is still obvious, to achieve the prescribed localization error, some local functions ($\tilde{\psi}_1, \tilde{\psi}_3$) have to extend along the high permeability channels and thus end up with relatively large supports. Recall that Remark 3.19 implies that the decay rate of $\psi$ and thus the radius of $\tilde{\psi}_i$ are invariant under local scaling (contrast scaling). But higher permeability means stronger connectivity and consequently larger patch sizes, and therefore $\tilde{\psi}_i$ extends farther in physical distance along high permeability channels. As a limitation of our approach, this long-range decaying compromises the sparsity of the localized basis and the stiffness matrix, which is an issue that we plan to resolve in our future work. As comparison, Figure 14 plots the log-profile and the supports of three localized functions $\tilde{\psi}_1^{\text{re}}, \tilde{\psi}_2^{\text{re}}, \tilde{\psi}_3^{\text{re}}$ obtained similarly but with the regular partition in the $200 \times 200$ resolution case. We can see that some localized functions under regular partition also have relative large support size. However, such long-distance extension is not the result of large patch sizes (since regular partition has uniform patch size) but the result of large condition factor $\delta(P, 1)$. Recall that to achieve an desired localization error, the radius of localized basis is also affected by $\log \delta(P, 1)$.

6. Multiresolution operator decomposition. One can see that what we have been doing with operator compression is essentially to truncate the microscopic/fine-scale part of $A$ while preserving the necessary macroscopic/coarse-scale part that dominates the accuracy. And in the meantime, the condition number of the compressed operator also drops to the level consistent to the prescribed accuracy. This consistency inspires us to perform the compression procedure hierarchically in order to separate the operator into multiple scales of resolution rather than just two.

Now consider that, instead of just compressing the inverse $A^{-1}$, we want to solve the problem $Ax = b$. Due to the sparsity of $A$, a straightforward idea is to employ iterative methods. But these methods will suffer from the large condition number of $A$. Alternatively, we would like to use the energy decomposition of $A$ and the locality
Fig. 13. Samples of the localized basis.

(i.e., sparsity) of the energy decomposition to resolve the difficulty of large condition number. The main idea is to decompose the computation of $A^{-1}$ into hierarchical resolutions such that (i) the relative condition number in each scale/level can be well bounded and (ii) the subsystem to be solved on each level is as sparse as the original $A$. Here we will make use of the choice of the partition $P$, the basis $\Phi$ and $\Psi$ obtained in sections 3 and 4, to serve the purpose of multiresolution operator decomposition. In the following, we first implement a one-level decomposition.

**6.1. One-level operator decomposition.** Let $P$, $\Phi$, $\Psi$, and $U$ be constructed as in Algorithm 1, namely,

(i) $P = \{P_j\}_{j=1}^M$ is a partition of $V$;

(ii) $\Phi = [\Phi_1, \Phi_2, \ldots, \Phi_M]$ such that every $\Phi_j \subset \text{span}\{P_j\}$ has dimension $q_j$;

(iii) $\Psi = A^{-1}\Phi(\Phi^T A^{-1}\Phi)^{-1}$;

(iv) $U = [U_1, U_2, \ldots, U_M]$ such that every $U_j \subset \text{span}\{P_j\}$ has dimension $\dim(\text{span}\{P_j\}) - q_j$ and satisfies $\Phi_j^TU_j = 0$.

Then, $[U, \Psi]$ forms a basis of $\mathbb{R}^n$, and we have

$$U^T A \Psi = U^T \Phi(\Phi^T A^{-1}\Phi)^{-1} = 0.$$
Thus, the inverse of $A$ can be written as

$$
A^{-1} = \left( \begin{bmatrix} U^T & \Psi^T \end{bmatrix}^{-1} \begin{bmatrix} U^T \\ \Psi^T \end{bmatrix} A \begin{bmatrix} U & \Psi \end{bmatrix} \begin{bmatrix} U \\ \Psi \end{bmatrix}^{-1} \right)^{-1} = U(U^T A U)^{-1} U^T + \Psi(\Psi^T A \Psi)^{-1} \Psi^T.
$$

In the following, we denote $\Psi^T A \Psi = A_{st}$ and $U^T A U = B_{st}$, respectively. We also use the phrase “solving $A^{-1}$” to mean “solving $A^{-1} b$ for any $b$.” From (66), we observe that solving $A^{-1}$ is equivalent to solving $A_{st}^{-1}$ and $B_{st}^{-1}$ separately. For $B_{st}$, notice that since the space/basis $U$ is constructed locally with respect to each patch $P_j$, $B_{st}$ will inherit the sparsity characteristic from $A$ if $A$ is local/sparse. Thus, it will be efficient to solve $B_{st}^{-1}$ using iterative-type methods if the condition number of $B_{st}$ is bounded.

In the following, we introduce Lemma 6.1, which provides an upper bounded of the $B_{st}$ that ensures the efficiency of solving $B_{st}^{-1}$. The proof of the lemma imitates the proof from Theorem 10.9 of [28], where the required condition (17) corresponds to equation (2.3) in [28].

**Lemma 6.1.** If $\Phi$ satisfies the condition (17) with constant $\epsilon$, then

$$
\lambda_{\text{max}}(B_{st}) \leq \lambda_{\text{max}}(A) \cdot \lambda_{\text{max}}(U^T U), \quad \lambda_{\text{min}}(B_{st}) \geq \frac{1}{\epsilon^2} \cdot \lambda_{\text{min}}(U^T U),
$$

and thus

$$
\kappa(B_{st}) \leq \epsilon^2 \cdot \lambda_{\text{max}}(A) \cdot \kappa(U^T U).
$$

**Proof.** For $\lambda_{\text{max}}(B_{st})$, we have

$$
\lambda_{\text{max}}(B_{st}) = \|B_{st}\|_2 = \|U^T A U\|_2 \leq \|A\|_2 \|U\|_2 \|U^T U\|_2 = \lambda_{\text{max}}(A) \lambda_{\text{max}}(U^T U).
$$

Fig. 14. Samples of the localized basis from regular partition.
For $\lambda_{\min}(B_{st})$, since $\Phi$ satisfies the condition (17) with constant $\epsilon$ and $\Phi^T U = 0$, we have

$$
\|x\|^2 \leq \frac{1}{\lambda_{\min}(U^T U)} x^T U^T U x \leq \frac{\epsilon^2}{\lambda_{\min}(U^T U)} x^T U^T A U x = \frac{\epsilon^2}{\lambda_{\min}(U^T U)} x^T B_{st} x,
$$

and thus $\lambda_{\min}(B_{st}) \geq \frac{1}{\epsilon^2} \lambda_{\min}(U^T U)$.

We shall have some discussion on the bound $\epsilon^2 \cdot \lambda_{\max}(A) \cdot \kappa(U^T U)$ separately into two parts, namely, (i) $\epsilon^2 \cdot \lambda_{\max}(A)$ and (ii) $\kappa(U^T U)$. Notice that $U^T U$ is actually block-diagonal with blocks $U_j^T U_j$; therefore,

$$
\kappa(U^T U) = \frac{\lambda_{\max}(U^T U)}{\lambda_{\min}(U^T U)} = \frac{\max_{1 \leq j \leq M} \lambda_{\max}(U_j^T U_j)}{\min_{1 \leq j \leq M} \lambda_{\min}(U_j^T U_j)}.
$$

In other words, we can bound $\kappa(U^T U)$ well by choosing proper $U_j$ for each $P_j$. For instance, if we allow any kind of local computation on $P_j$, we may simply extend $\Phi_j$ to an orthonormal basis of span{$P_j$} to get $U_j$ by using QR factorization [29]. In this case, we have $\kappa(U^T U) = 1$.

For the part $\epsilon^2 \lambda_{\max}(A)$, recall that we construct $\Phi$ based on a partition $P$ and an integer $q$ so that $\Phi$ satisfies condition (17) with constant $\epsilon(P,q)$; thus, the posterior bound of $\kappa(B_{st})$ is $\epsilon(P,q)^2 \lambda_{\max}(A)$ (when $\kappa(U^T U) = 1$). Recall that $\kappa(A_{st})$ is bounded by $\delta(P,q)\|A^{-1}\|_2$; therefore, $\kappa(A_{st}) \kappa(B_{st}) \leq \epsilon(P,q)^2 \delta(P,q) \kappa(A)$. That is, $A_{st}$ and $B_{st}$ divide the burden of the large condition number of $A$ with an amplification factor $\epsilon(P,q)^2 \delta(P,q)$. We call $\kappa(P,q) \triangleq \epsilon(P,q)^2 \delta(P,q)$, the $q$th-order condition number of the partition $P$. This explains why we attempt to bound $\kappa(P,q)$ in the construction of the partition $P$.

Ideally, we hope the one-level operator decomposition gives $\kappa(A_{st}) \approx \kappa(B_{st})$, so that the two parts equally share the burden in parallel. But such result may not be good enough when $\kappa(A_{st})$ and $\kappa(B_{st})$ are still large. To fully decompose the large condition number of $A$, a simple idea is to recursively apply the one-level decomposition. That is, we first set a small enough $\epsilon$ to sufficiently bound $\kappa(B_{st})$, then, if $\kappa(A_{st})$ is still large, we apply the decomposition to $A_{st}^{-1}$ again to further decompose $\kappa(A_{st})$. However, the decomposition of $A^{-1}$ is based on the construction of $P$ and $\Phi$, namely, on the underlying energy decomposition $E = \{E_k\}_{k=1}^m$ of $A$. Hence, we have to construct the corresponding energy decomposition of $A_{st}$ before we implement the same operator decomposition on $A_{st}^{-1}$.

**6.2. Inherited energy decomposition.** Let $E = \{E_k\}_{k=1}^m$ be the energy decomposition of $A$; then the **inherited energy decomposition** of $A_{st} = \Psi^T A \Psi$ with respect to $E$ is simply given by $E^\Psi = \{E_k^\Psi\}_{k=1}^m$, where

$$
E_k^\Psi = \Psi^T E_k \Psi, \quad k = 1, 2, \ldots, m.
$$

Notice that this inherited energy decomposition of $A_{st}$ with respect to $E$ has the same number of energy elements as $E$, which is not preferred and is actually redundant in practice. Therefore, we shall consider to reduce the energy decomposition of $A_{st}$. Indeed we will use $\Psi$ instead of $\Psi$ in practice, where each $\hat{\psi}_i$ is some local approximator of $\psi_i$ (obtained by Construction 2). Specifically, we will actually deal with $A_{st} = \hat{\Psi}^T A \hat{\Psi}$, and thus we shall consider to find a proper condensed energy decomposition of $A_{st}$. 
If we see $A_{st}$ as a matrix with respect to the reduced space $\mathbb{R}^N$, then for any vector $x = \{x_1, \ldots, x_N\} \in \mathbb{R}^N$, the connection $x \sim E\tilde{\Psi}$ between $x$ and some $E\tilde{\Psi} = \tilde{\Psi}^T E\tilde{\Psi}$ comes from the connection between $E$ and those $\tilde{\psi}_i$ corresponding to nonzero $x_i$, and such connections are the key to constructing a partition $P_{st}$ for $A_{st}$. Recall that the support of each $\tilde{\psi}_i(S_{st}(P_{st}))$ for some $k$) is a union of patches, so there is no need to distinguish among energy elements interior to the same patch when we deal with the connections between these elements and the basis $\tilde{\Psi}$. Therefore, we introduce the **reduced inherited energy decomposition** of $A_{st} = \tilde{\Psi}^T A\tilde{\Psi}$ as follows.

**Definition 6.2 (reduced inherited energy decomposition).** With respect to the underlying energy decomposition $E$ of $A$, the partition $P$, and the corresponding $\tilde{\Psi}$, the reduced inherited energy decomposition of $A_{st} = \tilde{\Psi}^T A\tilde{\Psi}$ is given by $\mathcal{E}_{st}^\Psi = \{A_{P_j}\}_{j=1}^M \cup \{E\tilde{\Psi} : E \in \mathcal{E}_{st}^P\}$ with

\begin{align}
\Delta_{P_j}^\Psi &= \tilde{\Psi}^T \Delta_{P_j} \tilde{\Psi}, \quad j = 1, 2, \ldots, M \\
E\tilde{\Psi} &= \tilde{\Psi}^T E\tilde{\Psi}, \quad \forall E \in \mathcal{E}_{st}^P
\end{align}

where $\mathcal{E}_{st}^P = \mathcal{E}\setminus\mathcal{E}_P$ with $\mathcal{E}_P = \{E \in \mathcal{E} : \exists P_j \in P \ s.t. \ E \in P_j\}$.

Once we have the underlying energy decomposition of $A_{st}$ (or $\tilde{A}_{st}$), we can repeat the procedure to decompose $A_{st}^{-1}$ (or $\tilde{A}_{st}^{-1}$) in $\mathbb{R}^n$ as what we have done to $A^{-1}$ in $\mathbb{R}^n$. We will introduce the **multilevel decomposition** of $A^{-1}$ in the following subsection.

**6.3. Multiresolution operator decomposition.** Let $A^{(0)} = A$, and we construct $A^{(k)}, B^{(k)}$ recursively from $A^{(0)}$. More precisely, let $\mathcal{E}^{(k-1)}$ be the underlying energy decomposition of $A^{(k-1)}$ and $P^{(k)}, \Phi^{(k)}, \Psi^{(k)}$ and $U^{(k)}$ be constructed corresponding to $A^{(k-1)}$ and $\mathcal{E}^{(k-1)}$ in space $\mathbb{R}^{N^{(k-1)}}$, where $N^{(k-1)}$ is the dimension of $A^{(k-1)}$. We use one-level operator decomposition to decompose $(A^{(k-1)})^{-1}$ as

\[
(A^{(k-1)})^{-1} = U^{(k)} \left( (U^{(k)})^T A^{(k-1)} U^{(k)} \right)^{-1} \left( (U^{(k)})^T + \Psi^{(k)} \right) \left( (\Psi^{(k)})^T A^{(k-1)} \Psi^{(k)} \right)^{-1} (\Psi^{(k)})^T
\]

and then define $A^{(k)} = (\Psi^{(k)})^T A^{(k-1)} \Psi^{(k)}, B^{(k)} = (U^{(k)})^T A^{(k-1)} U^{(k)},$ and $\mathcal{E}^{(k)} = (\mathcal{E}^{(k-1)})^{\Psi^{(k)}}_{re}$ as in Definition 6.2. Moreover, if we write

\begin{align}
\Phi^{(1)} &= \Phi^{(1)}, \quad \Phi^{(k)} = \Phi^{(1)} \Phi^{(2)} \cdots \Phi^{(k-1)} \Phi^{(k)}, \quad k \geq 1 \\
U^{(1)} &= U^{(1)}, \quad U^{(k)} = \Psi^{(1)} \Psi^{(2)} \cdots \Psi^{(k-1)} U^{(k)}, \quad k \geq 1 \\
\Psi^{(1)} &= \Psi^{(1)}, \quad \Psi^{(k)} = \Psi^{(1)} \Psi^{(2)} \cdots \Psi^{(k-1)} \Psi^{(k)}, \quad k \geq 1
\end{align}

then one can prove by induction that for $k \geq 1$,

\[
A^{(k)} = (\Psi^{(k)})^T A \Psi^{(k)} = \left( (\Phi^{(k)})^T A^{-1} \Phi^{(k)} \right)^{-1}, \quad B^{(k)} = (U^{(k)})^T A U^{(k)},
\]

\[
(\Phi^{(k)})^T \Phi^{(k)} = (\Phi^{(k)})^T \Psi^{(k)} = I_{N^{(k)}}, \quad \Psi^{(k)} = A^{-1} \Phi^{(k)} \left( (\Phi^{(k)})^T A^{-1} \Phi^{(k)} \right)^{-1},
\]

and that for any integer $K$,

\[
A^{-1} = (A^{(0)})^{-1} = \sum_{k=1}^K U^{(k)} \left( (U^{(k)})^T A U^{(k)} \right)^{-1} (U^{(k)})^T + \Psi^{(k)} \left( (\Psi^{(k)})^T A \Psi^{(k)} \right)^{-1} (\Psi^{(k)})^T.
\]
Remark 6.3.

- One shall notice that the partition $\mathcal{P}^{(k)}$ on each level $k$ is not a partition of the whole space $\mathbb{R}^n$ but a partition of the reduced space $\mathbb{R}^{N(k-1)}$ and that $\Phi^{(k)}$, $\Psi^{(k)}$, $U^{(k)}$ are all constructed corresponding to this $\mathcal{P}^{(k)}$ in the same reduced space. Intuitively, if the average patch size (basis number in a patch) for partition $\mathcal{P}^{(k)}$ is $s^{(k)}$, then we have $N(k) = \frac{q^{(k)}}{2\pi^2} N(k-1)$, where $q^{(k)}$ is the integer for constructing $\Phi^{(k)}$.

- Generally, methods of multiresolution type use nested partitions/meshes that are generated only based on the computational domain [3, 38]. But here the nested partitions are replaced by levelwisely constructed ones which are adaptive to $A^{(k)}$ on each level and require no a priori knowledge of the computational domain/space.

- In the gamblet setting introduced in [27], (74), together with (75), can be viewed as the gamblet transform.

The multiresolution operator decomposition up to a level $K$ is essentially equivalent to a decomposition of the whole space $\mathbb{R}^n$ [27] as

$$\mathbb{R}^n = U^{(1)} \oplus U^{(2)} \oplus \cdots \oplus U^{(K)} \oplus \Psi^{(K)},$$

where again we also use $U^{(k)}$ (or $\Psi^{(k)}$) to denote the subspace spanned by the basis $U^{(k)}$ (or $\Psi^{(k)}$). Due to the $A$-orthogonality between these subspaces, using this decomposition to solve $A^{-1}$ is equivalent to solving $A^{-1}$ in each subspace separately (or more precisely solving $(B^{(k)})^{-1}$, $k = 1, \ldots, K$, or $(A^{(k)})^{-1}$), and by doing so we decompose the large condition number of $A$ into bounded pieces as the following corollary states.

**Corollary 6.4.** If on each level $\Phi^{(k)}$ is given by Construction 1 with integer $q^{(k)}$, then for $k \geq 1$ we have

$$\lambda_{\max}(A^{(k)}) \leq \delta(\mathcal{P}^{(k)}, q^{(k)}), \quad \lambda_{\min}(A^{(k)}) \geq \lambda_{\min}(A),$$

$$\lambda_{\max}(B^{(k)}) \leq \delta(\mathcal{P}^{(k-1)}, q^{(k-1)}) \lambda_{\max}\left((U^{(k)})^T U^{(k)}\right),$$

$$\lambda_{\min}(B^{(k)}) \geq \frac{1}{\varepsilon(\mathcal{P}^{(k)}, q^{(k)})^2} \lambda_{\min}\left((U^{(k)})^T U^{(k)}\right),$$

and thus

$$\kappa(A^{(k)}) \leq \delta(\mathcal{P}^{(k)}, q^{(k)})\|A^{-1}\|_2,$$

$$\kappa(B^{(k)}) \leq \varepsilon(\mathcal{P}^{(k)}, q^{(k)})^2 \delta(\mathcal{P}^{(k-1)}, q^{(k-1)}) \kappa\left((U^{(k)})^T U^{(k)}\right).$$

For consistency, we write $\delta(\mathcal{P}^{(0)}, q^{(0)}) = \lambda_{\max}(A^{(0)}) = \lambda_{\max}(A)$.

**Proof.** These results follow directly from Theorem 3.22 and Lemma 6.1.

**Remark 6.5.** The fact $\lambda_{\min}(B^{(k)}) \geq \frac{1}{\varepsilon(\mathcal{P}^{(k)}, q^{(k)})^2}$ implies that the level $k$ is a level with resolution of scale no greater than $\varepsilon(\mathcal{P}^{(k)}, q^{(k)})$, namely, the space $U^{(k)}$ is a subspace of the whole space $\mathbb{R}^n$ of scale finer than $\varepsilon(\mathcal{P}^{(k)}, q^{(k)})$ with respect to $A$. This is essentially what multiresolution means in this decomposition.

Now that we have a multiresolution decomposition of $A^{-1}$, the applying of $A^{-1}$ (namely, solving linear system $Ax = b$) can break into the applying of $(B^{(k)})^{-1}$ on
each level and the applying of \((A^{(k)})^{-1}\) on the bottom level. In what follows, we always assume \(\kappa((U^{(k)})^T U^{(k)}) = 1\). Then the efficiency of the multiresolution decomposition in resolving the difficulty of large condition number of \(\mathbf{A}\) lies in the effort to bound each \(\varepsilon(\mathcal{P}^{(k)}, q^{(k)})^2 \delta(\mathcal{P}^{(k)}, q^{(k)})\) so that \(\mathcal{B}^{(k)}\) has a controlled spectrum width and can be efficiently solved using the CG-type method. Define \(\kappa(\mathcal{P}^{(k)}, q^{(k)}) = \varepsilon(\mathcal{P}^{(k)}, q^{(k)})^2 \delta(\mathcal{P}^{(k)}, q^{(k)})\) and \(\gamma^{(k)} = \frac{\varepsilon(\mathcal{P}^{(k)}, q^{(k)})}{\varepsilon(\mathcal{P}^{(k)}, q^{(k)})}\); then we can write

\[
\kappa(\mathcal{B}^{(k)}) \leq \varepsilon(\mathcal{P}^{(k)}, q^{(k)})^2 \delta(\mathcal{P}^{(k)}, q^{(k)}) = (\gamma^{(k)})^2 \kappa(\mathcal{P}^{(k)}, q^{(k)}).
\]

The partition condition number \(\kappa(\mathcal{P}^{(k)}, q^{(k)})\) is a levelwise information only concerning the partition \(\mathcal{P}^{(k)}\). Similar to what we do in Algorithm 4, we will impose a uniform bound \(c\) in the partitioning process so that \(\kappa(\mathcal{P}^{(k)}, q^{(k)}) \leq c\) on every level. The ratio \(\gamma^{(k)}\) reflects the scale gap between level \(k - 1\) and \(k\), which is why it should measure the condition number(spectrum width) of \(\mathcal{B}^{(k)}\). However, it turns out that the choice of \(\gamma^{(k)}\) is not arbitrary, and it will be subject to a restriction derived out of concern of sparsity.

So far the \(A^{(k)}\) and \(B^{(k+1)}\) are dense for \(k \geq 1\) since the basis \(\Psi^{(k)}\) is global. It would be pointless to bound the condition number of \(\mathcal{B}^{(k)}\) if we cannot take advantage of the locality/sparsity of \(\mathbf{A}\). So in practice, the multiresolution operator decomposition is performed with localization on each level to ensure locality/sparsity. Thus, we have the modified multiresolution operator decomposition in the following subsection.

### 6.4. Multiresolution operator decomposition with localization

Let \(\tilde{\mathbf{A}}^{(0)} = \mathbf{A}\), and we construct \(\tilde{\mathbf{A}}^{(k)}, \tilde{\mathcal{B}}^{(k)}\) recursively from \(\tilde{\mathbf{A}}^{(0)}\). More precisely, let \(\tilde{\mathcal{E}}^{(k-1)}\) be the underlying energy decomposition of \(\tilde{\mathbf{A}}^{(k-1)}\) and \(\mathcal{P}^{(k)}, \Phi^{(k)}, \Psi^{(k)}\) and \(U^{(k)}\) be constructed corresponding to \(\tilde{\mathbf{A}}^{(k-1)}\) and \(\tilde{\mathcal{E}}^{(k-1)}\) in space \(\mathbb{R}^{N^{(k-1)}}\). We decompose \((\tilde{\mathbf{A}}^{(k-1)})^{-1}\) as

\[
(\tilde{\mathbf{A}}^{(k-1)})^{-1} = U^{(k)} \left( (U^{(k)})^T \tilde{\mathbf{A}}^{(k-1)} U^{(k)} \right)^{-1} (U^{(k)})^T + \Psi^{(k)} \left( (\Psi^{(k)})^T \tilde{\mathbf{A}}^{(k-1)} \Psi^{(k)} \right)^{-1} (\Psi^{(k)})^T
\]

Let \(\tilde{\Psi}^{(k)}\) be a local approximator of \(\Psi^{(k)}\). Then we define

\[
\tilde{\mathbf{A}}^{(k)} = (\tilde{\Psi}^{(k)})^T \tilde{\mathbf{A}}^{(k-1)} \tilde{\Psi}^{(k)} + U^{(k)T} \tilde{\mathbf{A}}^{(k-1)} U^{(k)}, \quad \tilde{\mathcal{B}}^{(k)} = (U^{(k)})^T \tilde{\mathbf{A}}^{(k-1)} U^{(k)},
\]

and \(\tilde{\mathcal{E}}^{(k)} = (\tilde{\mathcal{E}}^{(k-1)})_{rer} \tilde{\Psi}^{(k)}\) as in Definition 6.2.

Similar to Corollary 6.4, we have the following estimates on the condition numbers of \(\tilde{\mathbf{A}}^{(k)}\) and \(\mathcal{B}^{(k)}\).

**Corollary 6.6.** If on each level \(\Phi^{(k)}\) is given by Construction 1 with integer \(q^{(k)}\) and \(\tilde{\Psi}^{(k)}\) is a local approximator of \(\Psi^{(k)}\) subject to localization error \(\|\tilde{\psi}_i^{(k)} - \psi_i^{(k)}\|_{A^{(k-1)}} \leq \frac{\epsilon}{\sqrt{\lambda_{\text{min}}}}\), then for \(k \geq 1\) we have

\[
\begin{align*}
\lambda_{\text{max}}(\tilde{\mathbf{A}}^{(k)}) &\leq \left(1 + \frac{\epsilon}{\sqrt{\delta(\mathcal{P}^{(k)}, q^{(k)})}}\right)^2 \delta(\mathcal{P}^{(k)}, q^{(k)}), \\
\lambda_{\text{min}}(\tilde{\mathbf{A}}^{(k)}) &\geq \lambda_{\text{min}}(\mathbf{A}), \\
\lambda_{\text{max}}(\tilde{\mathcal{B}}^{(k)}) &\leq \left(1 + \frac{\epsilon}{\sqrt{\delta(\mathcal{P}^{(k-1)}, q^{(k-1)})}}\right)^2 \delta(\mathcal{P}^{(k-1)}, q^{(k-1)}) \lambda_{\text{max}} \left( (U^{(k)})^T U^{(k)} \right), \\
\lambda_{\text{min}}(\tilde{\mathcal{B}}^{(k)}) &\geq \frac{1}{\varepsilon(\mathcal{P}^{(k)}, q^{(k)})^2} \lambda_{\text{min}} \left( (U^{(k)})^T U^{(k)} \right).
\end{align*}
\]
and thus
\[ \kappa(\tilde{A}^{(k)}) \leq \left(1 + \frac{\epsilon}{\sqrt{\delta(P^{(k)}, q^{(k)})}}\right)^2 \delta(P^{(k)}, q^{(k)})\|A^{-1}\|_2, \]
\[ \kappa(\tilde{B}^{(k)}) \leq \left(1 + \frac{\epsilon}{\sqrt{\delta(P^{(k)}, q^{(k-1)})}}\right)^2 \varepsilon(P^{(k)}, q^{(k)})^2 \delta(P^{(k-1)}, q^{(k-1)}) \kappa\left((U^{(k)})^T U^{(k)}\right). \]

For consistency, we write \( \delta(P^{(0)}, q^{(0)}) = \lambda_{\max}(\tilde{A}^{(0)}) = \lambda_{\max}(A) \).

**Proof.** These results follow directly from the proof of Theorem 3.22, Corollary 3.23, and Lemma 6.1.

One can indeed prove that \( \delta(P^{(k)}, q^{(k)}) \geq \frac{1}{\varepsilon(P^{(k)}, q^{(k)})} \), and thus \( \frac{\epsilon}{\sqrt{\delta(P^{(k)}, q^{(k)})}} \leq \varepsilon(P^{(k)}, q^{(k)}) \) is a small number. Therefore, Corollary 6.6 states that the multiresolution decomposition with localization has estimates on condition numbers of the same order as in Corollary 6.4, i.e., \( \kappa(\tilde{A}^{(k)}) \leq O(\varepsilon(P^{(k)}, q^{(k)})\|A^{-1}\|_2) \) and \( \kappa(\tilde{B}^{(k)}) \leq O(\varepsilon(P^{(k)}, q^{(k)})^2 \delta(P^{(k-1)}, q^{(k-1)})) \). Having this in hand, we proceed to discuss the desired sparsity of \( \tilde{A}^{(k)} \) and \( \tilde{B}^{(k)} \).

**Locality preservation.** Similar to the locality discussion of \( \tilde{A}_{st} \) in section 4, under the locality condition (56), we have the following recursive estimate on the number of nonzero entries of each \( A^{(k)} \):

\[ \text{nnz}(\tilde{A}^{(k)}) = O\left(\text{nnz}(\tilde{A}^{(k-1)}) \cdot (q^{(k)})^2 \cdot \frac{1}{s^{(k)}} \cdot (r^{(k)})^d\right), \]

where \( s^{(k)} \) is the average patch size of \( P^{(k)} \) and \( r^{(k)} \) is the decay radius of \( \tilde{\Psi}^{(k)} \). Also, noticing that \( \tilde{B}^{(k)} = (U^{(k)})^T \tilde{A}^{(k-1)} U^{(k)} \) and that the basis \( U^{(k)} \) are local vectors of support size \( s^{(k)} \), we have

\[ \text{nnz}(\tilde{B}^{(k)}) = O(\text{nnz}(\tilde{A}^{(k-1)}) \cdot s^{(k)}). \]

In fact, the basis \( U^{(k)} \) can be computed from \( \Phi^{(k)} \) using the implicit QR factorization [29], and thus the matrix multiplication with respect to \( U^{(k)} \) can be done by using the Householder vectors in time linear to \( q^{(k)} \cdot N^{(k)} \). Therefore, when we evaluate \( \tilde{B}^{(k)} = (U^{(k)})^T \tilde{A}^{(k-1)} U^{(k)} \) (in the iterative method), only the NNZ of \( \tilde{A}^{(k-1)} \) matters. In brief, we need to preserve the locality of \( A^{(k)} \) down through all the levels to ensure the efficiency of the multiresolution decomposition with localization. But the accumulation of the factor \( (q^{(k)})^2 (s^{(k)})^d \), if not well controlled, will compromise the sparsity inherited from \( \tilde{A}^{(0)} = A \). Therefore, a necessary condition for the decomposition to keep sparsity is

\[ o(s^{(k)}) \geq (q^{(k)})^2 (r^{(k)})^d, \quad k \geq 1, \]

under which we have the sparsity estimate \( \text{nnz}(\tilde{A}^{(k)}) = O(\text{nnz}(A)) \). In particular, when we impose the localization error \( \|\tilde{\psi}_i^{(k)} - \tilde{\psi}_i^{(k)}\|_{A^{(k-1)}} = \frac{\epsilon}{\sqrt{N^{(k)}}} \), on each level \( k \) for some uniform \( \epsilon \), we have \( r^{(k)} = O(\log \frac{1}{\epsilon} + \log N^{(k)} + \log \delta(P^{(k)}, q^{(k)})) \) according the discussions in subsection 3.2. Then the sparsity condition becomes

\[ o(s^{(k)}) \geq (q^{(k)})^2 \left(\log \frac{1}{\epsilon} + \log N^{(k)} + \log \delta(P^{(k)}, q^{(k)})\right)^d, \quad k \geq 1. \]

This lower bound of the patch size \( s^{(k)} \) means that we need to compress enough dimensions from a higher level to a lower level in order to preserve sparsity due to the outreaching support of the localized basis \( \tilde{\Psi}^{(k)} \).
In practice, we will choose some \( \epsilon \) smaller than the top-level scale \( \varepsilon(\mathcal{P}^{(1)}, q^{(1)}) \) and a uniform integer \( q \). By imposing uniform condition bound \( \kappa(\mathcal{P}^{(k)}, q^{(k)}) \leq c \), we have \( \delta(\mathcal{P}^{(k)}, q^{(k)}) \leq \frac{\varepsilon(\mathcal{P}^{(k)}, q^{(k)})}{\varepsilon(\mathcal{P}^{(k)}, q^{(k)})^2} \leq \frac{c}{s^2} \). Therefore, a safe uniform criterion for patch size \( s^{(k)} \) is
\[
O(s^{(k)}) = s = q^2 \left( \log \frac{1}{\epsilon} + \log n \right)^{d+l}
\]
for some small \( l > 0 \), which asymptotically, when \( n \) goes large and \( \epsilon \) goes small, will ensure \( \text{nnz}(A^{(k)}) = O(\text{nnz}(A)) \) down through the decomposition. Since the decomposition should stop when \( N^{(K)} \), the dimension of \( A^{(K)} \) is small enough; namely, when \( n = O((s/q)^K) \), (82) also gives us an estimate of the total level number as
\[
K = O(\log_{s/q} n) = O \left( \frac{\log n}{\log(q(\log \frac{1}{\epsilon} + \log n)^{d+l})} \right) = O \left( \frac{\log n}{\log(\log(\log \frac{1}{\epsilon} + \log n))} \right).
\]

**Choice of scale ratio \( \gamma \).** Recall that the partition \( \mathcal{P}^{(k)} \) is a partition of basis in the space \( \mathbb{R}^{N^{(k-1)}} \). By tracing back to the top level, we can also see it as a partition in the original space \( \mathbb{R}^n \). Denoting \( R^{(k)} \) to be the average radius (with respect to adjacency defined by \( A \)) and \( S^{(k)} \) to be the average patch size of the patches (with respect to \( \mathbb{R}^n \)) in \( \mathcal{P}^{(k)} \), we have \( S^{(k)} = O((R^{(k)})^d) \) under the locality condition (56), and an intuitive geometry estimate gives \( S^{(k)} \). As a consequence, under the local energy decomposition condition (57) of order \( (q,p) \), we have the following estimate:
\[
\gamma^{(k)} = \frac{\varepsilon(\mathcal{P}^{(k)}, q^{(k)})}{\varepsilon(\mathcal{P}^{(k-1)}, q^{(k-1)})} = O \left( \left( \frac{R^{(k)}}{R^{(k-1)}} \right)^p \right) = O \left( \left( \frac{S^{(k)}}{S^{(k-1)}} \right)^{\frac{p}{q}} \right) = O \left( (s^{(k)})^{\frac{p}{q}} \right).
\]
Such estimate arises naturally in a lot of PDE problems, especially when the smallest eigenvalues of local operators have clear dependence on the domain size, the dimension of the space, and the order of the equation [16]. Under the sparsity condition (81) and considering \( q \) as a constant, we require
\[
o(\gamma^{(k)}) \geq \left( \log \frac{1}{\epsilon} + \log N^{(k)} + \log \delta \left( \mathcal{P}^{(k)}, q^{(k)} \right) \right)^p
\]
to ensure the sparsity of the decomposition, and similarly, a safe, uniform choice of the scale ratio \( \gamma^{(k)} \) is
\[
\gamma^{(k)} = \gamma = \left( \log \frac{1}{\epsilon} + \log n \right)^{p+l}
\]
for some small \( l > 0 \). Such choice provides a uniform bound on the condition number of \( \tilde{B}^{(k)} \) as
\[
\kappa(\tilde{B}^{(k)}) \leq O \left( \frac{\varepsilon(\mathcal{P}^{(k)}, q^{(k)})^2 \delta(M^{(k)}_r, q^{(k-1)})}{\varepsilon(\mathcal{P}^{(k)}, q^{(k-1)})} \right) \leq O \left( \left( \log \frac{1}{\epsilon} + \log n \right)^{p+l} \right)
\]
when a uniform condition bound \( \kappa(\mathcal{P}^{(k)}, q^{(k)}) \leq c \) is imposed by algorithm. Notice that the ratio \( \gamma^{(k)} \) is only defined for \( k \geq 2 \); thus, the estimate (87) is valid for \( k \geq 2 \). For consistency, we choose \( \varepsilon(\mathcal{P}^{(1)}, q^{(1)})^2 = O\left( \frac{(\log \frac{1}{\epsilon} + \log n)^{p+l}}{\|A\|_2} \right) \) so that (87) is also valid for \( k = 1 \).

**Remark 6.7.** By estimate (87), the bound on \( \kappa(\tilde{B}^{(k)}) \) will go to infinity when \( n \) goes to infinity. In our construction of the multiresolution decomposition for resolving
a large condition number of $A$, we cannot asymptotically have an absolute constant bound for $\kappa(\tilde{B}^{(k)})$ on all levels due to the required preservation of sparsity. This difficulty comes from the inductive nature of the algorithm that the posterior estimate of the sparsity of $\tilde{A}^{(k)}$ is based on the sparsity of $\tilde{A}^{(k-1)}$, as shown in (79). However, in [28], the existence of nested measurement function $\Phi$ is assumed a priori before the construction of the multiresolution structure, and thus the sparsity of $\tilde{A}^{(k)}$ can be inherited directly from $\tilde{A}^{(0)}$, which avoids the accumulation of the factor $\frac{q^{(k)}(\epsilon^{(k)})^k}{k}$
through levels. As a result, the sparsity of $\tilde{A}^{(k)}$ does not contradict the uniform bound of $\kappa(\tilde{B}^{(k)})$.

**Error estimate.** Using multiresolution operator decomposition with localization to solve $A^{-1}$, the error on each level comes from two main sources: (i) the localization error between $\tilde{\Psi}^{(k)}((\tilde{\Psi}^{(k)})^T A^{(k-1)} \tilde{\psi}^{(k)})^{-1}((\tilde{\Psi}^{(k)})^T)$ and $\tilde{\Psi}^{(k)}((\tilde{\Psi}^{(k)})^T A^{(k-1)} \tilde{\psi}^{(k)})^{-1}((\tilde{\Psi}^{(k)})^T)$ and (ii) the error caused by solving $(A^{(k)})^{-1}((\tilde{\Psi}^{(k)})^T A^{(k-1)} \tilde{\psi}^{(k)})^{-1}((\tilde{\Psi}^{(k)})^T)$ with iterative-type methods. To come up with an estimate of the total error, we perform a standard analysis of error accumulation in an inductive manner.

**Theorem 6.8.** Given an integer $K$, let $\text{Inv}(A)$ denote the solver for $A^{-1}$ using $K$-level’s multiresolution operator decomposition with localization. Assume that

(i) each $\tilde{B}^{(k)}$ can be solved efficiently subject to a uniform relative error bound $\text{err}_B$ in the sense that the solver $\text{Inv}(\tilde{B}^{(k)})$ (as a linear operator) satisfies

$$\|(\tilde{B}^{(k)})^{-1} - \text{Inv}(\tilde{B}^{(k)})\|_{\tilde{B}^{(k)}} \leq \text{err}_B \|b\|_{\tilde{B}^{(k)}}^{-1} \quad \forall b \in \mathbb{R}^{N^{(k)}} \quad \forall 1 \leq k \leq K;$$

(ii) at level $K$, $(\tilde{A}^{(K)})^{-1}$ can be solved efficiently subject to a relative error $\text{err}_{A}^{(K)}$ in the sense that the solver $\text{Inv}(\tilde{A}^{(K)})$ satisfies

$$\|(\tilde{A}^{(K)})^{-1} - \text{Inv}(\tilde{A}^{(K)})\|_{\tilde{A}^{(K)}} \leq \text{err}_{A}^{(K)} \|b\|_{\tilde{A}^{(K)}}^{-1} \quad \forall b \in \mathbb{R}^{N^{(K)}};$$

(iii) each $\tilde{\psi}^{(k)}$ satisfies the localization approximation property

$$\|\tilde{\psi}_{i}^{(k)} - \psi_{i}^{(k)}\|_{\tilde{A}^{(k-1)}} \leq \text{err}_{\text{loc}} \frac{\text{err}_{\text{loc}}}{2\sqrt{N^{(k)}}\|A^{-1}\|_2}, \quad 1 \leq i \leq N^{(k)};$$

with a uniform constant $\text{err}_{\text{loc}}$. Then we have

$$\|A^{-1}b - \text{Inv}(A)b\|_A \leq \text{err}_{\text{total}} \|b\|_{A^{-1}} \quad \forall b \in \mathbb{R}^n,$$

and in $\|\cdot\|_2$,

$$\|A^{-1}b - \text{Inv}(A)b\|_2 \leq \text{err}_{\text{total}} \|A^{-1}\|_2 \|b\|_2 \quad \forall b \in \mathbb{R}^n,$$

where

$$\text{err}_{\text{total}} = K(\text{err}_B + \text{err}_{\text{loc}}) + \text{err}_{A}^{(K)}.$$

**Remark 6.9.** Assumption 6.8 is reasonable since each $\tilde{B}^{(k)}$ inherits the sparsity from $A$, and its condition numbers can be well bounded in order $O((\log\text{err}_{\text{loc}}^{-1} + \log n)^{p+1})$. Assumption 6.8 is reasonable since each $\tilde{A}^{(K)}$ is of small dimension when $K$ is as large as in (83). Assumption 6.8 is reasonable due to the exponential decay property of each $\tilde{\psi}^{(k)}$. Indeed, to ensure locality of reduced energy decomposition, the localization error control can be relaxed in practice. Such relaxed error can be fixed by doing compensation computation as we will see in subsection 6.7.
6.5. Algorithm. We now summarize the procedure of MMD with localization as Algorithm 6 and the use of MMD to solve linear system as Algorithm 7. Also, Figure 15 shows the flowchart of Algorithm 6.

Algorithm 6. MMD with Localization.

**Input:** SPD matrix $A = \tilde{A}(0)$, energy decomposition $E = \tilde{E}(0)$, underlying basis $V$, localization constant $\epsilon$, level number $K$, $q(k)$, error factor bound $\varepsilon(k)$ and condition bound $c(k)$ for each level.

**Output:** $\tilde{A}(K)$, $\tilde{\Psi}(k)$, $U(k)$, and $\tilde{B}(k)$.

1: for $k = 1 : K$ do
2: Construct $P(k), \Phi(k), U(k), \tilde{\Psi}(k)$ with Algorithm 1, with respect to $\tilde{A}(k-1), \tilde{E}(k-1)$, and subject to $q(k), \varepsilon(k), c(k)$ and localization error $\sqrt{\frac{\epsilon}{N(k)}}$;
3: Compute $\tilde{A}(k)$ and $\tilde{B}(k)$ by Equation (78);
4: Compute reduced energy $\tilde{E}(k)$ by Equation (73);
5: output/store $\tilde{\Psi}(k), U(k), \tilde{B}(k)$;
6: end for
7: output/store $\tilde{A}(K)$.

Algorithm 7. Solving linear system with MMD with Localization.

**Input:** $\tilde{\Psi}(k), U(k), \tilde{B}(k)$ for $k = 1, 2, \ldots, K$, $\tilde{A}(K)$, load vector $b = b(0)$, prescribed relative accuracy $\varepsilon$

**Output:** Approximated solution $x(0)$.

1: for $k = 1 : K$ do
2: $z(k) = (U(k))^T b(k-1)$;
3: Solve $\tilde{B}(k) y(k) = z(k)$ up to relative error $\varepsilon$;
4: $b(k) = (\tilde{\Psi}(k))^T b(k-1)$;
5: end for
6: Solve $\tilde{A}(K) x(K) = b(K)$ up to relative error $\varepsilon$;
7: for $k = K : 1$ do
8: $x(k-1) = U(k) y(k) + \tilde{\Psi}(k) x(k)$;
9: end for

Fig. 15. Process flowchart of Algorithm 6.

Remark 6.10.
- Once the MMD is obtained, the first for-loop (line 1) in Algorithm 7 can be performed in parallel, which makes it much more efficient than nonparallelizable iterative methods.
- Once the whole decomposition structure is completed, we can a posteriori omit the levelwise energy decompositions and partitions. Then if we see our levelwisely constructed $\Phi$ as a nested sequence (74a), our decomposition is structurally equivalent to the result obtained in [28], where the existence of
such nested $\Phi$ is a priori assumed. Therefore, the required properties of the
nested sequence in Condition 2.3 of [28] are similar to the assumption in
Theorem 6.8.

Complexity of Algorithm 6. Assume that locality conditions (56), (57), (58)
are true with constant $d, p, q, c$. Then all $\varepsilon^{(k)}$ and $c^{(k)}$ are chosen uniformly over levels
to be $q, c$, respectively. $\varepsilon^{(k)}$ is chosen subject to scale ratio choice (86), $c^{(1)} = (\log 1 + \log n)^{\frac{p+1}{2}}$
for some small $l > 0$, and $\epsilon$ is chosen so that $\epsilon \leq \varepsilon^{(1)}$. Due to the
condition bound $c$, we have condition number estimate (87). Then the complexity of
line 2 can be modified from (62) as

$$O \left( d \cdot s^2 \cdot \log s \cdot n \right) + O(\log s \cdot n \cdot \log n)
+ O \left( q \cdot n \cdot \left( \log \frac{1}{\epsilon} + \log n \right)^{p+l} \cdot \left( \log \frac{1}{\epsilon} + \log n \right)^{d+1} \right),$$

where $s = O((\log 1 + \log n)^{d(1+l/p)})$ according to estimate (84). The complexity of
lines 3 and 4 (sparse matrices multiplication) together can be bounded by

$$O \left( n \cdot \left( \log \frac{1}{\epsilon} + \log n \right)^{3d} \right)$$

due to the locality of $\tilde{\Psi}^{(k)}$ and the inherited locality of $\tilde{A}^{(k-1)}$ and $\tilde{B}^{(k-1)}$. Therefore,
the complexity on each level can be bounded by

$$O \left( d \cdot s^2 \cdot \log s \cdot n \right) + O(\log s \cdot n \cdot \log n) + O \left( q \cdot n \cdot \left( \log \frac{1}{\epsilon} + \log n \right)^{3d+p} \right),$$

where we have assumed that $d \geq 1 \geq l$. Then the total complexity of Algorithm 6 is
the level number $K$ times (92). By (83), we have $K = O\left( \frac{\log n}{\log(\log \frac{1}{\epsilon} + \log n)} \right) \leq O(\log n)$
and $K \log s = O(\log n)$. Thus, the total complexity of Algorithm 6 is

$$O \left( d \cdot s^2 \cdot \log s \cdot n \cdot \log n \right) + O(n \cdot (\log n)^2) + O \left( K \cdot q \cdot n \cdot \left( \log \frac{1}{\epsilon} + \log n \right)^{3d+p} \right)$$

$$\leq O(m \cdot \log n \cdot (\log \frac{1}{\epsilon} + \log n)^{3d+p}),$$

where $m = O(d \cdot n)$ is the number of nonzero entries of $A$.

Complexity of Algorithm 7. Assume that the relative accuracy $\epsilon$ is the same
as the $\epsilon$ in Algorithm 6. Recall that the number of nonzero entries of each $\tilde{A}^{(k)}$ is
bounded by $O(n\text{nz}(A)) = O(m)$, and that the condition number of each $\tilde{B}^{(k)}$ can be
bounded by $O((\log \frac{1}{\epsilon} + \log n)^{p+1})$; then the complexity of solving the linear system in
line 3 using a CG-type method is bounded by

$$O \left( m \cdot \left( \log \frac{1}{\epsilon} + \log n \right)^{p+l} \cdot \log \frac{1}{\epsilon} \right).$$

Therefore, if we use a CG-type method to solve all inverse problems involved in
Algorithm 7, based on the MMD with localization given by Algorithm 6, the running
time of Algorithm 7 subject to levelwise relative accuracy $\epsilon$ is
Intuitively, this approximation lies above the scale of $\varepsilon$ the following theorem.

\[
K \text{over all levels finer than level } \| (95) \text{ integer not to the whole space } \mathbb{R}^n \text{ error factor the A constructed level by level using the information of } \Phi \text{ notice that the composite basis should have a corresponding dominant compression error. However, we should again } \varepsilon \text{ However, by Theorem 6.8, the total accuracy is } \varepsilon_{\text{total}} = O(K\varepsilon). \text{ Thus, the complexity of Algorithm 7 subject to a total relative accuracy } \varepsilon_{\text{total}} \text{ is }
\]

\[
(93) \quad O \left( m \cdot \left( \log \frac{1}{\varepsilon_{\text{total}}} + \log n \right)^{p+l} \cdot \log \frac{1}{\varepsilon_{\text{total}}} \right) \leq O \left( m \cdot \left( \log \frac{1}{\varepsilon} + \log n \right)^{p+l} \cdot \log \frac{1}{\varepsilon} \cdot \log n \right).
\]

6.6. Multilevel operator compression. Also, we can consider the MMD from the perspective of operator compression. For any $K$, by omitting the finer scale subspaces $U(k)$, $k = 1, 2, \ldots, K$, we get an effective approximator of $A^{-1}$ as

\[
A^{-1} \approx \Psi(K) \left( (\Psi(K))^T A \Psi(K) \right)^{-1} (\Psi(K))^T = P_{\Psi(K)}^A A^{-1}.
\]

Intuitively, this approximation lies above the scale of $\varepsilon(\mathcal{P}(k), q(k))$ and therefore should have a corresponding dominant compression error. However, we should again notice that the composite basis $\Phi(k)$ is not given a priori and directly in $\mathbb{R}^n$ but constructed level by level using the information of $A(k)$ on each level and recall that the error factor $\varepsilon(\mathcal{P}(k), q(k))$ is computed with respect to the reduced space $\mathbb{R}^N(k)$, not to the whole space $\mathbb{R}^n$. Thus, the total error of compression (94) is accumulated over all levels finer than level $K$. To quantify such compression error, we introduce the following theorem.

Theorem 6.11. Assume that on each level $\Phi(k)$ is given by Construction 1 with integer $q(k)$. Then we have

\[
(95) \quad \| x - P_{\Phi(k)} x \|_2 \leq \left( \sum_{k=0}^{K} \varepsilon(\mathcal{P}(k), q(k))^2 \right)^{\frac{1}{2}} \| x \|_A \quad \forall x \in \mathbb{R}^n,
\]

and thus for any $x \in \mathbb{R}^n$ and $b = Ax$, we have

\[
\| x - P_{\Phi(k)}^A x \|_A \leq \left( \sum_{k=1}^{K} \varepsilon(\mathcal{P}(k), q(k))^2 \right)^{\frac{1}{2}} \| b \|_2,
\]

\[
\| x - P_{\Phi(k)}^A x \|_2 \leq \left( \sum_{k=1}^{K} \varepsilon(\mathcal{P}(k), q(k))^2 \right) \| b \|_2,
\]

\[
\| A^{-1} - P_{\Phi(k)}^A A^{-1} \|_2 \leq \left( \sum_{k=1}^{K} \varepsilon(\mathcal{P}(k), q(k))^2 \right).
\]

Remark 6.12.

• Though the compression error is in a cumulative form, if we assume that $\varepsilon(\mathcal{P}(k), q(k))$ increases with $k$ at a certain ratio $\frac{\varepsilon(\mathcal{P}(k), q(k))}{\varepsilon(\mathcal{P}(k-1), q(k-1))} \geq \gamma$ for some $\gamma > 1$, then it is easy to see that

\[
\sum_{k=1}^{K} \varepsilon(\mathcal{P}(k), q(k))^2 \leq \frac{\gamma^2}{\gamma^2 - 1} \varepsilon(\mathcal{P}(k), q(k))^2,
\]

which is an error of scale $\varepsilon(\mathcal{P}(k), q(k))^2$ as we expected.
Again one shall be aware of the difference between the one-level compression with error factor $\epsilon(P(K),q(K))$ and the multilevel compression in subsection 6.3. A one-level compression with error factor $\epsilon(P(K),q(K))$ requires constructing $P(K), \Phi(k)$ and so on directly with respect to $A$ in the whole space $\mathbb{R}^n$, which involves solving eigenvalue problems on considerably large patches in $P(K)$ when $\epsilon(P(K),q(K))$ is a coarse scale. But the multilevel compression in subsection 6.3 is computed hierarchically with bounded compression ratio between levels and thus only involves eigenvalue problems on patches of well-bounded size ($s = O(\log \frac{1}{\epsilon} + \log n)^{d+1}$) in each reduced space $\mathbb{R}^{N(k)}$ and is thus more tractable in practice.

One can also analyze the compression error when localization of each $\Psi(k)$ is considered. The analysis would be similar to the one in Theorem 6.11.

6.7. Numerical example for MMD. Our third numerical example shows the effectiveness of using MMD with localization to solve a graph Laplacian system. Again we use the same setup in Example 1 in subsection 5.1 with density factor $\eta = 2$. But this time the vertices of the graph are randomly distributed around a 2-D roll surface of area 1 in $\mathbb{R}^3$. The distribution is a combination of a uniform distribution over the surface and up to a 10% random displacement off the surface. More precisely, the 2-D roll is characterized as

$$\begin{align*}
(x(t), y(t), z) &= (\rho(t) \cos(\theta(t)), \rho(t) \sin(\theta(t)), z), \\
\theta(t) &= \frac{1}{a} \log (1 + t (e^{4\pi a} - 1)), \\
\rho(t) &= \frac{a}{\sqrt{1 + a^2}} \left( t + \frac{1}{e^{4\pi a} - 1} \right),
\end{align*}$$

and so $\sqrt{(\rho(t))'^2 + (\rho(t)\theta(t))'^2} = 1$. Each vertex $(x_i, y_i, z_i)$ is generated by

$$\begin{align*}
(x_i, y_i, z_i) &= (\eta_i \rho(t_i) \cos(\theta(t_i)), \eta_i \rho(t_i) \sin(\theta(t_i)), z_i), \\
&\text{where } t_i \sim_{i.i.d} U[0, 1], \\
&z_i \sim_{i.i.d} U[0, 1], \text{ and } \eta_i \sim_{i.i.d} U[0.9, 1.1].
\end{align*}$$

In this example, we take $n = 10000$ and $a = 0.1$. Figure 16(a) and 16(b) shows the point cloud of all vertices. This explicit expression, however, is considered as hidden geometric information and is not employed in our partitioning algorithm.

The Laplacian $L = A_0$ and the energy decomposition $E$ are given as in Example 2.10, and we apply a 4-level MMD with localization using Algorithm 6 to decompose the problem of solving $L^{-1}$. In this particular case we have $\lambda_{\max}(L) = 1.93 \times 10^7$.

![Front view](a) Front view
![Top view](b) Top view

**Fig. 16.** “Roll surface” constructed by (96).
and \( \lambda_{\min}(L) = 1 \). Again for graph Laplacian, we choose \( q = 1 \). On each level \( k \), the partition is constructed subject to \( \varepsilon(\mathcal{P}(k), 1) \leq 10^{-6} \) (i.e., \( \{ \varepsilon(\mathcal{P}(k), 1) \}^4_{k=1} = \{ 0.00001, 0.0001, 0.001, 0.01 \} \)) and \( \delta(\mathcal{P}(k), 1) \varepsilon(\mathcal{P}(k), 1)^2 \leq 50 \). The compliment space \( U^{(k)} \) is extended from \( \Phi^{(k)} \) using a patchwise QR factorization. So according to Corollary 6.6, each \( \kappa(B^{(k)}), k = 2, 3, 4 \) is expected to be bounded by \( \delta(\mathcal{P}(k-1), 1) \varepsilon(\mathcal{P}(k), 1)^2 = \delta(\mathcal{P}(k-1), 1) \varepsilon(\mathcal{P}(k), 1) \varepsilon(\mathcal{P}(k), 1)^2 \leq 500 \). \( \kappa(B^{(1)}) \) is expected to be bounded by \( \lambda_{\max}(L) \varepsilon(\mathcal{P}(0), 1)^2 \leq 1.93 \times 10^2 \), and \( \kappa(A^{(4)}) \) is expected to be bounded by \( \delta(\mathcal{P}(3), 1) \lambda_{\min}(L)^{-1} = \delta(\mathcal{P}(3), 1) \varepsilon(\mathcal{P}(3), 1) \varepsilon(\mathcal{P}(3), 1)^2 \lambda_{\min}(L)^{-1} \leq 5000 \). Since we will use a CG-type method to compare the effectiveness of solving \( L^{-1} \) directly and using the 4-level decomposition, the complexities of both approaches are proportional to the product of the NNZ entries and the condition number of the matrix concerned, given a fixed prescribed relative accuracy [29]. Therefore, we define the complexity of a matrix as the product of its NNZ entries and its condition number. Though here we use the sparsity of \( B^{(k)} = (U^{(k)})^T A^{(k-1)} U^{(k)} \), in practice only the sparsity of \( A^{(k-1)} \) matters, and the matrix multiplication with respect to \( U^{(k)} \) can be done by using the Householder vectors from the implicit QR factorization [29]. The results not only satisfy the theoretical prediction but also turn out to be much better than expected, as shown in Table 2 and Figure 17. We now verify the performance of the 4-level decomposition by solving two particular systems \( Lu^* = b \), where

Case 1: \( u^*_i = (x^2_i + y^2_i + z^2_i)^{\frac{1}{2}}, i = 1, 2, \ldots, n, b = Lu^* \);

Case 2: \( u^*_i = x_i + y_i + \sin(z_i), i = 1, 2, \ldots, n, b = Lu^* \).
from the “Main Cost 5-level” columns of Tables 4 and 5, we can see that this simple cases are mainly concentrated on level 2, namely, the inverting of \( \tilde{B}^{(3)} \). This observation implies that the structural/geometrical details of \( L \) have more proportion on the scale corresponding to level 2 than on other scales, which is consistent with the fact that \( \tilde{B}^{(3)} \) on level 2 has the largest complexity of all. Though in practice we do not have the information in Tables 4 and 5, we may observe the dominance of \( \tilde{B}^{(3)} \) on level 2 after numbers of calls of our solver. As a natural improvement, we can simply further decompose the problem at level 2. More precisely, to relieve the dominance of level 2, we add one extra scale of 0.0003 between the scales of 0.0001 and 0.001. Consequently, we obtain a similar 5-level decomposition with \( \{\varepsilon(P^{(k)}, 1)^2\}_{k=1}^5 \) = \{0.00001, 0.0001, 0.0003, 0.001, 0.01\} (see Table 3 and Figure 18). From the “Main Cost 5-level” columns of Tables 4 and 5, we can see that this simple

| Level | Size | #Nonzero | Condition Number | Complexity |
|-------|------|----------|------------------|------------|
| \( L \) | 10000 × 10000 | 128018 \( \hat{=} m \) | 1.93 × 10^7 | 2.47 × 10^{12} |
| \( \tilde{B}^{(1)} \) | 1898 × 1898 | 9812 \( \approx 0.07m \) | 1.72 × 10^2 | 1.69 × 10^6 |
| \( \tilde{B}^{(2)} \) | 6639 × 6639 | 391499 \( \approx 3.06m \) | 1.80 × 10^1 | 7.04 × 10^6 |
| \( \tilde{B}^{(3)} \) | 1014 × 1014 | 237212 \( \approx 1.85m \) | 2.56 × 10^1 | 6.09 × 10^6 |
| \( \tilde{B}^{(4)} \) | 313 × 313 | 86323 \( \approx 0.67m \) | 1.62 × 10^1 | 1.40 × 10^6 |
| \( \tilde{B}^{(5)} \) | 114 × 114 | 12996 \( \approx 0.10m \) | 5.54 × 10^1 | 7.20 × 10^5 |
| \( \tilde{A}^{(5)} \) | 22 × 22 | 442 \( \approx 0.004m \) | 1.60 × 10^3 | 7.07 × 10^5 |
| total | - | 738284 \( \approx 5.77m \) | - | 1.76 × 10^7 |
Table 4

Case 1: Difference between 4-level and 5-level decomposition.

| Level | Scale                  | Matrix | # Iteration | Main Cost   |
|-------|------------------------|--------|-------------|-------------|
|       | 1/\lambda_{\text{max}}(L) \sim 1/\lambda_{\text{min}}(L) |        |             |             |
| 0     | $10^{-5} \sim 1/\lambda_{\text{max}}(L)$ | $\bar{B}(1)$ | 22          | $2.15 \times 10^5$ |
|       | $10^{-5} \sim 10^{-4}$ | $\bar{B}(1)$ | 22          | $2.15 \times 10^5$ |
| 1     | $10^{-5} \sim 10^{-4}$ | $\bar{B}(2)$ | 11          | $4.31 \times 10^6$ |
|       | $10^{-5} \sim 10^{-4}$ | $\bar{B}(2)$ | 11          | $4.31 \times 10^6$ |
| 2     | $10^{-4} \sim 10^{-3}$ | $\bar{B}(3)$ | 25          | $3.32 \times 10^6$ |
|       | $10^{-4} \sim 3 \times 10^{-3}$ | $\bar{B}(3)$ | 14          | $1.21 \times 10^6$ |
| 3     | $10^{-3} \sim 10^{-2}$ | $\bar{B}(4)$ | 23          | $2.86 \times 10^5$ |
|       | $3 \times 10^{-3} \sim 10^{-2}$ | $\bar{B}(4)$ | 22          | $2.86 \times 10^5$ |
| 4     | $10^{-2} \sim 1/\lambda_{\text{min}}(L)$ | $\bar{A}(4)$ | 31          | $9.72 \times 10^3$ |
|       | $10^{-2} \sim 1/\lambda_{\text{min}}(L)$ | $\bar{A}(5)$ | 22          | $9.72 \times 10^3$ |
| 0     | -                      | L      | 25          | $3.84 \times 10^6$ |
| Total | -                      | -      | 137         | $1.90 \times 10^7$ |
| Parallel | -                  | -      | 135         | $1.31 \times 10^7$ |
|        |                        |        |             | $1.36 \times 10^7$ |
|        |                        |        |             | $8.15 \times 10^6$ |
Table 5

Case 2: Difference between 4-level and 5-level decomposition.

| Level | Scale | Matrix | # Iteration | Main Cost |
|-------|-------|--------|-------------|-----------|
|       | 1/λ_{max}(L) \sim 1/λ_{min}(L) | L | 586 | 7.50 \times 10^7 |
| 0     | 10^{-5} \sim 1/λ_{max}(L) | \tilde{B}^{(1)} | 24 | 2.35 \times 10^5 |
|       | 10^{-5} \sim 10^{-4} | \tilde{B}^{(2)} | 11 | 4.31 \times 10^6 |
| 1     | 10^{-3} \sim 10^{-2} | \tilde{B}^{(3)} | 25 | 1.04 \times 10^7 |
|       | 10^{-2} \sim 1/λ_{min}(L) | \tilde{A}^{(4)} | 14 | 1.21 \times 10^6 |
| 2     | 3 \times 10^{-3} \sim 10^{-3} | \tilde{B}^{(5)} | 23 | 7.96 \times 10^5 |
|       | 10^{-3} \sim 10^{-2} | \tilde{A}^{(5)} | 22 | 3.08 \times 10^4 |
| 3     | 10^{-2} \sim 1/λ_{min}(L) | L | 16 | 2.05 \times 10^6 |
|       | 10^{-2} \sim 10^{-1} | L | 23 | 2.94 \times 10^6 |
| Total | - | - | 129 | 1.78 \times 10^7 |
| Parallel | - | - | 131 | 1.23 \times 10^7 |
|        | - | - | - | 1.24 \times 10^7 |
|        | - | - | - | 7.25 \times 10^6 |
improvement of further decomposition based on the feedback of computational results does reduce the computational cost.

7. Conclusion and future works.

7.1. Summary. In this work, we introduce the notion of energy decomposition for SPD matrix $A$ using the representation of energy elements. These energy elements help extract the hidden geometric information of the operator, which serves the purpose of finding an appropriate partitioning of the basis. Specifically, we introduce the closed and interior energies which tightly bound the restricted energy/restricted matrix in terms of positive definiteness. These bounds further lead into the introduction of two important local measurements, the error factor and the condition factor, which can be calculated efficiently by solving a local and partial eigenproblem. Using these local measurements, we propose a nearly-linear time algorithm to obtain an appropriate basis partitioning for compressing the operator with prescribed accuracy and bounded condition number. Extending the idea of operator compression into hierarchical formulation, we also propose a nearly linear time solver for a general SPD matrix. The main idea is to decompose the operator into multiple scales of resolution such that the relative condition number in each scale can be bounded. Experimental results are reported to demonstrate the efficacy of our proposed algorithms.

7.2. Future works. This groundwork introduces the idea of energy decomposition and its applications in operator compression and solving SPD linear systems. We believe that the energy framework may prompt further research. Particularly, we discover further possible improvement of our algorithms during the development stage.

Due to the pairing characteristic of Algorithm 4, we are quite affirmative that our partitioning algorithm is not optimal. Instead, the clustering problem could be reformulated into some local optimization problem such that the construction of the partition $P$ can be more robust. Second, our current implementation is a combination of Matlab and C++ coding, and no parallel computing is included. Therefore, one of our future works is to develop an optimal coding such that more comparison experiments with state-of-the-art algorithms can be conducted.

In applicationwise planning, we observe in subsection 5.2 that the compression scheme not only satisfies the prescribed accuracy and well-posedness requirement of the operator but also preserves the geometric information of the spectrum. More specifically, we recall that the eigenvectors corresponding to the first few eigenvectors of the compressed operator $A_{st}$ give accurate approximation of the corresponding eigenvectors of the original inverse operator $A^{-1}$ (see Figure 11). Inspired by this observation, we place confidence in modifying our compression scheme into an efficient
solver for a partial eigenproblem. Second, as demonstrated in the high-contrast problem in subsection 5.2, we believe that our energy decomposition framework can be specifically modified to suit the purpose of solving elliptic PDEs with high-contrast coefficients. Based on our energy decomposition, more in-depth analysis and improvement could be made to show that such a framework is one of the possible candidates to solve the elliptic-type problem with highly varying coefficients. Third, regarding the partitioning procedure and the locality of the basis in our algorithms, the localized MMD solver can be further improved to fit the needs of frequent updating of the solver. For example, in a graph Laplacian system, our MMD solver can be updated dynamically if new vertices/new edges are added to the given graphs. This dynamic update greatly reduces the time for the regeneration of the solver, especially when the update size is small.

Appendix: Proofs of Theorems.

Proof of Lemma 3.1.
1. Let $y = A^{-1}(P_\Phi b) \in \Psi$; then

$$\|x - y\|_A^2 = (x - y, A(x - y)) = (x - y - P_\Phi(x - y), b - P_\Phi b) \leq \|x - y - P_\Phi(x - y)\|_2 \|b - P_\Phi b\|_2 \leq \epsilon \|x - y\|_A \|b\|_2,$$

and thus we have

$$\|x - P_\Phi^A x\|_A \leq \|x - y\|_A \leq \epsilon \|b\|_2.$$

2. Let $z = A^{-1}(x - P_\Phi^A x)$; then

$$\|x - P_\Phi^A x\|_2^2 = (x - P_\Phi^A x, x - P_\Phi^A x) = (x - P_\Phi^A x, Az) = (x - P_\Phi^A x, z - P_\Phi^A z)_A \leq \|x - P_\Phi^A x\|_A \|z - P_\Phi^A z\|_A \leq \epsilon \|x - P_\Phi^A x\|_A \|Az\|_2,$$

and thus

$$\|x - P_\Phi^A x\|_2 \leq \epsilon \|x - P_\Phi^A x\|_A \leq \epsilon^2 \|b\|_2.$$

3. Immediate result of 2. □

Proof of Lemma 3.4. Let $\Phi^k \subset \text{span}\{S\}$ denote the eigenspace of $A_S$ (as an operator restricted to $\text{span}\{S\}$) corresponding to interior eigenvalues $\lambda_1(S) \leq \lambda_2(S) \leq \cdots \leq \lambda_k(S)$. On the one hand, for all $x \in \text{span}\{S\}$, we have

$$\|x\|_{A_S}^2 \geq \|x - P_{\Phi^{(i)}} x\|_{A_S}^2 = (x - P_{\Phi^{(i)}} x)^T A_S (x - P_{\Phi^{(i)}} x) \geq \lambda_{q(e)}(S) \|x - P_{\Phi^{(i)}} x\|_2^2,$$

$$\Rightarrow \|x - P_{\Phi^{(i)}} x\|_2 \leq \frac{1}{\sqrt{\lambda_{q(e)}(S)}} \|x\|_{A_S} \leq \epsilon \|x\|_{A_S}.$$

Thus, $\Phi^{(i)} \in \mathcal{G}(\epsilon)$, $q(e) \geq p(e)$. On the other hand, assume that the minimum $p(e)$ is achieved by some space $\Theta$; then one can check that
\[ \lambda_{p(\epsilon)+1} = \max_{\Theta \in \text{span}\{S\}} \min_{x \in \text{span}\{S\}, \dim \Theta = p(\epsilon)} \frac{\|x - P_\Theta x\|^2_{A_\delta}}{\|x - P_{\Theta_0} x\|^2_2} \]

\[ \geq \min_{x \in \text{span}\{S\}} \frac{\|x - P_\Theta x\|^2_{A_\delta}}{\|x - P_{\Theta_0} x\|^2_2} = \min_{y = x - P_{\Theta_0} x, \quad x \in \text{span}\{S\}} \frac{\|y\|^2_{A_\delta}}{\|y - P_{\Theta_0} y\|^2_2} \]

\[ = \min_{y = x - P_{\Theta_0} x, \quad x \in \text{span}\{S\}} \frac{\|y\|^2_{A_\delta}}{\|y - P_{\Theta_0} y\|^2_2} \geq \frac{1}{\epsilon^2}, \]

which implies \( p(\epsilon) \geq q(\epsilon) \) by the definition of \( q(\epsilon) \). Finally, we have \( p(\epsilon) = q(\epsilon) \).

**Proof of Lemma 3.6.** We only need to prove property 1; properties 2 and 3 follow by using the same argument as in Lemma 3.1. Recall that we have

\[ \|x - P_\Psi^T A x\|_2 = \|x - \Psi c\|_2 \leq \epsilon \|b\|_2 \]

with \( c = A_{st}^{-1} \Psi^T A x \). Let \( y_1 = \Psi c = \sum_{i=1}^N c_i \psi_i, \ y_2 = \tilde{\Psi} c = \sum_{i=1}^N c_i \tilde{\psi}_i \). Then we have

\[ \|y_1 - y_2\|_2 = \left\| \sum_{i=1}^N c_i (\psi_i - \tilde{\psi}_i) \right\|_2 \leq \left\| \sum_{i=1}^N c_i \|\psi_i - \tilde{\psi}_i\|_A \right\| \leq \frac{C \sqrt{N}}{\sqrt{N}} \left( \sum_{i=1}^N c_i^2 \right)^{\frac{1}{2}} = C \epsilon \sqrt{\epsilon} c^T e. \]

Notice that

\[ c^T c = x^T A \Psi A_{st}^{-2} \Psi^T A x \leq \|A_{st}^2 \Psi A_{st}^{-2} \Psi^T A^2\|_2 \|x\|^2_A, \]

\[ \|A_{st}^2 \Psi A_{st}^{-2} \Psi^T A^2\|_2 = \|A_{st}^{-1} \Psi^T A \Psi A_{st}^{-1}\|_2 = \|A_{st}^{-1}\|_2, \]

\[ \|x\|^2_A = b^T A^{-1} b \leq \|A^{-1}\|_2 \|b\|^2_2; \]

therefore, we get

\[ \|y_1 - y_2\|_2 \leq C \sqrt{\epsilon} \|A_{st}^{-1}\|_2 \|A^{-1}\|_2 \|b\|^2_2 \leq C \epsilon \|A^{-1}\|_2 \|b\|^2_2. \]

Then we have

\[ \|x - y_2\|_2 \leq \|x - y_1\|_2 + \|y_1 - y_2\|_2 \leq \epsilon \|b\|^2_2 + C \epsilon \|A^{-1}\|_2 \|b\|^2_2 = (1 + C \|A^{-1}\|_2) \epsilon \|b\|^2_2. \]

Since \( y_2 \in \text{span}\{\tilde{\Psi}\} \), we obtain

\[ \|x - P_\Psi^T A x\|_2 \leq \|x - y_2\|_2 \leq (1 + C \|A^{-1}\|_2) \epsilon \|b\|^2_2. \]

**Proof of Theorem 3.16.** \( \|\psi_i\|_A \leq \|\psi^k_i\|_A \leq \|\psi^0_i\|_A \) has been proved in the construction of local approximators. We only need to prove \( \|\psi^0_i\|_A \leq \sqrt{\delta(P_{\lambda_i})} \). Recall that \( \psi^0_i \) is defined as

\[ \psi^0_i = \arg \min_{x \in \text{span}(P_{\lambda_i})} \|x\|_A, \]

subject to \( \varphi_{i'}^T x = \delta_{i',i} \quad \forall \ i' \in \{1, \ldots, N\} \).
Without loss of generality, we can assume that \( \varphi_i \) is the first column of \( \Phi_j \). And notice that \( \|x\|_A = \|x\|_{A_{P_j}} \); therefore, the optimization formation can be rewritten as

\[
\psi_i^0 = \arg \min_{x \in \text{span}(P_j)} \|x\|_{A_{P_j}}
\]

(97) subject to \( \Phi_j^T x = z_i \),

where \( z_i = (1, 0, 0, \ldots, 0)^T \in \mathbb{R}^q_j \). This optimization problem can be uniquely and explicitly solved as

\[
\psi_i^0 = A_{P_j}^{-1} \Phi_j (\Phi_j^T A_{P_j}^{-1} \Phi_j)^{-1} z_i,
\]

(98) where again \( A_{P_j}^{-1} \) denotes the inverse of \( A_{P_j} \) as an operator restricted to \( \text{span}\{P_j\} \).

And thus we have

\[
\|\psi_i^0\|_A^2 = \|\psi_i^0\|_{A_{P_j}}^2 = z_i^T (\Phi_j^T A_{P_j}^{-1} \Phi_j)^{-1} z_i.
\]

Notice that

\[
\overline{A}_{P_j} \geq A_{P_j} \Rightarrow A_{P_j}^{-1} \geq \overline{A}_{P_j}^{-1} \Rightarrow \Phi_j^T A_{P_j}^{-1} \Phi_j \geq \overline{\Phi}_j^T \overline{A}_{P_j}^{-1} \Phi_j
\]

\[
\Rightarrow (\overline{\Phi}_j^T \overline{A}_{P_j}^{-1} \Phi_j)^{-1} \geq (\Phi_j^T A_{P_j}^{-1} \Phi_j)^{-1}
\]

since \( \overline{A}_{P_j} \) and \( A_{P_j} \) are both SPD as operators restricted to \( \text{span}\{P_j\} \). Therefore, by (34), we have

\[
\|\psi_i^0\|_A^2 \leq z_i^T (\Phi_j^T \overline{A}_{P_j}^{-1} \Phi_j)^{-1} z_i \leq \|\overline{\Phi}_j^T \overline{A}_{P_j}^{-1} \Phi_j\|_2 \|z_i\|_2^2 = \delta(P_j).
\]

Proof of Corollary 3.20. This proof basically follows the idea in [27]. For any \( \psi_i \), recall that \( \psi_i^k \in \text{span}(S_k(P_j)) \); thus, \( \overline{A}_{S_k(P_j)} \psi_i^k = 0 \) and

\[
\|\psi_i^k\|_{\overline{A}_{S_k}(P_j)}^2 = \|\psi_i - \psi_i^k\|_{\overline{A}_{S_k}(P_j)}^2 \leq \|\psi_i - \psi_i^k\|_A^2 \leq \left( \frac{\alpha(P)}{\alpha(P)} \right)^k \delta(P_j).
\]

For any two \( \psi_i, \psi_i' \), since \( (\psi_i - \psi_i')^T \Phi = 0 \) and \( A_{S_k} \psi_i' \in \Phi \), we have \( (\psi_i - \psi_i')^T A_{S_k} \psi_i' = 0 \) for all \( k \). Also notice that \( (\psi_i^k)^T A_{S_k} \psi_i^k = 0 \) for \( k < \frac{k_{ij}}{2} \) since \( S_k(P_j) \cap S_k(P_{j'}) = \emptyset \) when \( 2k < k_{ij} \). Therefore, taking \( k = \left\lfloor \frac{k_{ij}}{2} \right\rfloor - 1 \), we have

\[
|\psi_i^T A_{S_k} \psi_i'| = |(\psi_i^k)^T A(\psi_i' - \psi_i^k)| \leq \|\psi_i^k\|_A \|\psi_i' - \psi_i^k\|_A
\]

\[
\leq \left\lfloor \frac{\alpha(P)}{\alpha(P)} \right\rfloor^k \delta(P) \leq \left( \frac{\alpha(P)}{\alpha(P)} \right)^{\frac{k_{ij}}{2} - 1} \delta(P).
\]

Proof of Theorem 3.22. Thanks to (24) and since \( \Phi^T \Phi = I_M \), we have

\[
\|A^{-1}\|_2 = \|\Phi^T A^{-1} \Phi\|_2 \leq \|A^{-1}\|_2 \implies \lambda_{\min}(A_{st}) \geq \lambda_{\min}(A).
\]

Thanks to (34) and since \( A \preceq \sum_{j=1}^M \overline{A}_{P_j} \), we have

\[
\delta(P) I_M \geq (\Phi^T \sum_{j=1}^M \overline{A}_{P_j})^{-1} = A_{st} \implies \lambda_{\min}(A_{st}) \leq \delta(P).
\]
Proof of Corollary 3.23. Notice that $P^A_\Psi = \tilde{\Psi} A^{-1}_s \tilde{\Psi}^T A$ is a projection with respect to the energy inner product; we have

$$A \tilde{\Psi} A^{-1}_s \tilde{\Psi}^T A \preceq A \quad \implies \quad \tilde{\Psi} A^{-1}_s \tilde{\Psi}^T A \preceq A^{-1},$$

and since $\Phi^T \tilde{\Psi} = I_N$, we have

$$\|A^{-1}_s\|_2 \leq \|\Phi^T A^{-1}_s \Phi\|_2 \leq \|A^{-1}\|_2 \quad \implies \quad \lambda_{\min}(A_s) \geq \lambda_{\min}(A).$$

For any $c \in \mathbb{R}^N$, using a similar argument in Lemma 3.6, we have

$$\|\Psi c - \tilde{\Psi} c\|_A \leq C\epsilon \|c\|_2;$$

then we get

$$c^T A_s c = \|\tilde{\Psi} c\|^2 \leq \left(\|\Psi c - \tilde{\Psi} c\|_A + \|\Psi c\|_A\right)^2 \leq (C\epsilon + \sqrt{\lambda_{\max}(A_s)^2})^2 \|c\|^2_2;$$

and using $\lambda_{\max}(A_s) \leq \delta(P)$, we have

$$\lambda_{\max}(\tilde{A}_s) \leq \left(1 + \frac{C\epsilon}{\delta(P)}\right)^2 \delta(P).$$

Proof of Theorem 6.8. First, by Assumption 6.8, we have

$$\|(\tilde{A}^{(k)})^{-1} b - \operatorname{Inv}(\tilde{A}^{(k)}) b\|_{\tilde{A}^{(k)}} \leq \text{err}_{A}^{(k)} \|b\|_{(\tilde{A}^{(k)})^{-1}} \quad \forall b \in \mathbb{R}^{N^{(k)}}.$$

To perform induction, we assume that at level $k$, $(\tilde{A}^{(k)})^{-1}$ can be solved subject to a relative error err$^{(k)}_A$ in the sense that the solver Inv$(\tilde{A}^{(k)})$ satisfies

$$\|(\tilde{A}^{(k)})^{-1} b - \operatorname{Inv}(\tilde{A}^{(k)}) b\|_{\tilde{A}^{(k)}} \leq \text{err}_{A}^{(k)} \|b\|_{(\tilde{A}^{(k)})^{-1}} \quad \forall b \in \mathbb{R}^{N^{(k)}}.$$

Recall that $(\tilde{A}^{(k-1)})^{-1}$ and the solver Inv$(\tilde{A}^{(k-1)})$ are given by

$$\begin{align*}
(\tilde{A}^{(k-1)})^{-1} &= U^{(k)} \left((U^{(k)})^T \tilde{A}^{(k-1)} U^{(k)}\right)^{-1} (U^{(k)})^T \\
&\quad + \tilde{\Psi}^{(k)} \left((\tilde{\Psi}^{(k)})^T \tilde{A}^{(k-1)} \tilde{\Psi}^{(k)}\right)^{-1} (\tilde{\Psi}^{(k)})^T \\
\text{(103)}
\end{align*}$$

and

$$\text{Inv}(\tilde{A}^{(k-1)}) = U^{(k)} \text{Inv}(\tilde{B}^{(k)})(U^{(k)})^T + \tilde{\Psi}^{(k)} \text{Inv}(\tilde{A}^{(k)})(\tilde{\Psi}^{(k)})^T.$$

Then for any $b \in \mathbb{R}^{N^{(k-1)}}$, we have

$$\begin{align*}
\|(\tilde{A}^{(k-1)})^{-1} b - \operatorname{Inv}(\tilde{A}^{(k)}) b\|_{\tilde{A}^{(k-1)}} &\leq \|U^{(k)} \left((U^{(k)})^T \tilde{A}^{(k-1)} U^{(k)}\right)^{-1} (U^{(k)})^T b - U^{(k)} \text{Inv}(\tilde{B}^{(k)})(U^{(k)})^T b\|_{\tilde{A}^{(k-1)}} \\
&\quad + \|\tilde{\Psi}^{(k)} \left((\tilde{\Psi}^{(k)})^T \tilde{A}^{(k-1)} \tilde{\Psi}^{(k)}\right)^{-1} (\tilde{\Psi}^{(k)})^T b\|_{\tilde{A}^{(k-1)}} \\
&\quad - \|\tilde{\Psi}^{(k)} \left((\tilde{\Psi}^{(k)})^T \tilde{A}^{(k-1)} \tilde{\Psi}^{(k)}\right)^{-1} (\tilde{\Psi}^{(k)})^T b\|_{\tilde{A}^{(k-1)}} \\
&\quad + \|\tilde{\Psi}^{(k)} \left((\tilde{\Psi}^{(k)})^T \tilde{A}^{(k-1)} \tilde{\Psi}^{(k)}\right)^{-1} (\tilde{\Psi}^{(k)})^T b - \tilde{\Psi}^{(k)} \text{Inv}(\tilde{A}^{(k)})(\tilde{\Psi}^{(k)})^T b\|_{\tilde{A}^{(k-1)}} \\
&= I_1 + I_2 + I_3.
\end{align*}$$
Recall that $\tilde{A}^{(k)} = (\tilde{\Psi}^{(k)})^T \tilde{A}^{(k-1)} \tilde{\Psi}^{(k)}$ and $\tilde{B}^{(k)} = (U^{(k)})^T \tilde{A}^{(k-1)} U^{(k)}$; then by Assumption 6.8, we have

\[
I_1 = \|((\tilde{B}^{(k)})^{-1} (U^{(k)})^T b - \text{Inv}(\tilde{B}^{(k)})(U^{(k)})^T b\|_{\tilde{B}^{(k)}}
\leq \text{err}_B \left( b^T U^{(k)} (\tilde{B}^{(k)})^{-1} (U^{(k)})^T b \right)^{\frac{1}{2}} \\
\leq \text{err}_B \|((\tilde{A}^{(k-1)})^\frac{1}{2} U^{(k)} ((U^{(k)})^T \tilde{A}^{(k-1)} U^{(k)})^{-1} (U^{(k)})^T (\tilde{A}^{(k-1)})^\frac{1}{2} \| b \|_{\tilde{A}^{(k-1)}}^{-1}) \leq \text{err}_B \| b \|_{\tilde{A}^{(k-1)}}^{-1}.
\]

Similarly, by the assumption of induction, we have

\[
I_3 = \|((\tilde{A}^{(k)})^{-1} (\tilde{\Psi}^{(k)})^T b - \text{Inv}(\tilde{A}^{(k)})(\tilde{\Psi}^{(k)})^T b\|_{\tilde{A}^{(k)}} \leq \text{err}_A \| b \|_{\tilde{A}^{(k-1)}}^{-1}.
\]

Let $x = (\tilde{A}^{(k-1)})^{-1} b$; then we get

\[
I_2 = ||P_{\tilde{\Psi}^{(k)}}^{\tilde{A}^{(k-1)}} x - P_{\tilde{\Psi}^{(k)}}^{\tilde{A}^{(k-1)}} P_{\tilde{\Psi}^{(k)}}^{\tilde{A}^{(k-1)}} x||_{\tilde{A}^{(k-1)}} \\
\leq ||P_{\tilde{\Psi}^{(k)}}^{\tilde{A}^{(k-1)}} x - P_{\tilde{\Psi}^{(k)}}^{\tilde{A}^{(k-1)}} P_{\tilde{\Psi}^{(k)}}^{\tilde{A}^{(k-1)}} x||_{\tilde{A}^{(k-1)}} + ||P_{\tilde{\Psi}^{(k)}}^{\tilde{A}^{(k-1)}} P_{\tilde{\Psi}^{(k)}}^{\tilde{A}^{(k-1)}} x||_{\tilde{A}^{(k-1)}}.
\]

Using a similar argument in Lemma 3.6, we can actually prove by Assumption 6.8 that

\[
||P_{\tilde{\Psi}^{(k)}}^{\tilde{A}^{(k-1)}} x - P_{\tilde{\Psi}^{(k)}}^{\tilde{A}^{(k-1)}} P_{\tilde{\Psi}^{(k)}}^{\tilde{A}^{(k-1)}} x||_{\tilde{A}^{(k-1)}} \leq \frac{1}{2} \text{err}_{\text{loc}} ||P_{\tilde{\Psi}^{(k)}}^{\tilde{A}^{(k-1)}} x||_{\tilde{A}^{(k-1)}} \leq \frac{1}{2} \text{err}_{\text{loc}} || b ||_{\tilde{A}^{(k-1)}}^{-1}
\]

and

\[
||P_{\tilde{\Psi}^{(k)}}^{\tilde{A}^{(k-1)}} P_{U^{(k)}}^{\tilde{A}^{(k-1)}} x||_{\tilde{A}^{(k-1)}} \\
\leq \|((\tilde{\Psi}^{(k)})^T \tilde{A}^{(k-1)} \tilde{\Psi}^{(k)})^{-1}\|_2 \|((\tilde{\Psi}^{(k)})^T \tilde{A}^{(k-1)} P_{U^{(k)}}^{\tilde{A}^{(k-1)}} x\|_2 \\
\leq \|A^{-1}\|_2 \|((\tilde{\Psi}^{(k)} - \Psi^{(k)})^T \tilde{A}^{(k-1)} P_{U^{(k)}}^{\tilde{A}^{(k-1)}} x\|_2 \\
\leq \|A^{-1}\|_2 \|((\tilde{\Psi}^{(k)} - \Psi^{(k)})^T \tilde{A}^{(k-1)} (\tilde{\Psi}^{(k)} - \Psi^{(k)}))\|_2 \| P_{U^{(k)}}^{\tilde{A}^{(k-1)}} x\|_{\tilde{A}^{(k-1)}} \\
\leq \frac{1}{2} \text{err}_{\text{loc}} || b ||_{\tilde{A}^{(k-1)}}^{-1};
\]

thus, $I_2 \leq \text{err}_{\text{loc}} || b ||_{\tilde{A}^{(k-1)}}^{-1}$. Finally, we have

\[
||((\tilde{A}^{(k-1)})^{-1} b - \text{Inv}(\tilde{A}^{(k-1)})b||_{\tilde{A}^{(k-1)}} \leq (\text{err}_B + \text{err}_{\text{loc}} + \text{err}_A^{(k)}) || b ||_{\tilde{A}^{(k-1)}}^{-1};
\]

that is, we have

\[
\text{err}_A^{(k-1)} = (\text{err}_B + \text{err}_{\text{loc}} + \text{err}_A^{(k)}).
\]

Then by induction, the relative total error using $K$-level’s decomposition with localization for solving $A^{-1}$ is

\[
\text{err}_{\text{total}} = \text{err}_A^{(0)} = K(\text{err}_B + \text{err}_{\text{loc}}) + \text{err}_A^{(K)}.
\]

Proof of Theorem 6.11. Again by Lemma 3.1, we only need to prove (95). For consistency, we write $\Phi^{(0)} = I_n$ and correspondingly $\tilde{\Psi}^{(0)} = I_n$, $P_{\Phi^{(0)}} = I_n$, $P_{\Phi^{(0)}} = I_n$. Using (74), it is easy to check that for any $x \in \mathbb{R}^n$ and any $k_1 \leq k_2 \leq k_3$,

\[
(P_{\Phi^{(k_1)}} x - P_{\Phi^{(k_2)}} x)^T (P_{\Phi^{(k_2)}} x - P_{\Phi^{(k_3)}} x) = 0;
\]
thus, we have
\[ \|x - P_{\Phi(k)}x\|_2^2 = \left\| \sum_{k=1}^{K} (P_{\Phi(k-1)}x - P_{\Phi(k)}x) \right\|_2^2 \]
\[ = \sum_{k=1}^{K} \|P_{\Phi(k-1)}x - P_{\Phi(k)}x\|_2^2 \]
\[ = \sum_{k=1}^{K} \|P_{\Phi(k-1)}x - P_{\Phi(k)}x\|_2^2. \]

Notice that
\[ P_{\Phi(k-1)}x - P_{\Phi(k)}x = \Phi(k-1)(\Phi(k-1)^T x - \Phi(k-1)\Phi(k)^T (\Phi(k-1)^T x); \]
thus, by the construction of \( \Phi(k) \) (or \( \Phi(k) \)), we have
\[ \|P_{\Phi(k-1)}x - P_{\Phi(k)}x\|_2^2 \]
\[ = \|((\Phi(k-1)^T x - \Phi(k)^T (\Phi(k-1)^T x)|^2 \]
\[ \leq \varepsilon(P(k), q(k))^2 \|x\|^2_A \]
\[ = \varepsilon(P(k), q(k))^2 \|x\|^2_A \]
\[ = \varepsilon(P(k), q(k))^2 \|x\|^2_A. \]

We have used the fact that
\[ \|P_{\Psi(k)}x\|_2^2 = x^T A \Psi(k)^T (\Phi(k))^T A \Psi(k)^T A x \]
\[ = x^T \Phi(k)^T A^{-1} \Phi(k)^T A x \quad \forall k \geq 0. \]

Therefore, we have
\[ \|x - P_{\Phi(k)}x\|_2^2 \leq \sum_{k=0}^{K} \varepsilon(P(k), q(k))^2 \|x\|_A^2. \]

Acknowledgment. We would like to thank Prof. Houman Owhadi for the suggestions and discussions throughout the development of this work.

REFERENCES

[1] J. Ashburner, A fast diffeomorphic image registration algorithm, Neuroimage, 38 (2007), pp. 95–113.
[2] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. Van der Vorst, Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, SIAM, Philadelphia, PA, 1994.
[3] K.-J. Bathe and E. L. Wilson, Numerical Methods in Finite Element Analysis, Vol. 197, Prentice Hall, Englewood Cliffs, NJ, 1976.
[4] M. Belkin and P. Niyogi, Laplacian eigenmaps and spectral techniques for embedding and clustering, NIPS, 14 (2001), pp. 585–591.
[5] M. Belkin, P. Niyogi, and V. Sindhwani, Manifold regularization: A geometric framework for learning from labeled and unlabeled examples, J. Mach. Learn. Res., 7 (2006), pp. 2399–2434.
[6] F. L. Bookstein, Principal warps: Thin-plate splines and the decomposition of deformations, IEEE Trans. Pattern Anal. Mach. Intell., 11 (1989), pp. 567–585.
[7] Y. Boykov and V. Kolmogorov, Computing geodesics and minimal surfaces via graph cuts, ICCV, 3 (2003), pp. 26–33.
[8] S. C. Brenner and C. Carstensen, *Finite element methods*, in Encyclopedia of Computational Mechanics, Vol. 1, E. Stein, R. de Borst, and T. J. R. Hughes, eds., John Wiley, Chichester, 2004, p. 4.

[9] D. Cai, X. He, J. Han, and T. S. Huang, *Graph regularized nonnegative matrix factorization for data representation*, IEEE Trans. Pattern Anal. Mach. Intell., 33 (2011), pp. 1548–1560.

[10] Y. Cao, M. I. Miller, R. L. Winslow, and L. Younes, *Large deformation diffeomorphic metric mapping of vector fields*, IEEE Trans. Med. Imaging, 24 (2005), pp. 1216–1230.

[11] T. F. Chan and J. Zou, *Additive Schwarz domain decomposition methods for elliptic problems on unstructured meshes*, Numer. Algorithms, 8 (1994), pp. 329–346.

[12] F. R. Chung, *Spectral Graph Theory*, Vol. 92, American Mathematical Society, Providence, RI, 1997.

[13] C. Colovos and T. O. Yeates, *Verification of protein structures: Patterns of nonbonded atomic interactions*, Protein Sci., 2 (1993), pp. 1511–1519.

[14] A. d’Aspremont, L. El Ghaoui, M. I. Jordan, and G. R. Lanckriet, *A direct formulation for sparse PCA using semidefinite programming*, SIAM Rev., 49 (2007), pp. 434–448.

[15] W. Hackbusch, *Multi-Grid Methods and Applications*, Vol. 4, Springer Science & Business Media, New York, 2013.

[16] Y. T. Hou and P. Zhang, *Sparse operator compression of higher-order elliptic operators with rough coefficients*, Res. Math. Sci., 4 (2017), p. 24, http://doi.org/10.1186/s40687-017-0113-1.

[17] I. Jolliffe, *Principal Component Analysis*, Wiley Online Library, New York, 2002.

[18] R. Kolluri, J. R. Shewchuk, and J. F. O’Brien, *Spectral surface reconstruction from noisy point clouds*, in Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing, ACM, 2004, pp. 11–21.

[19] I. Koutis, G. L. Miller, and R. Peng, *A nearly-m log n time solver for SDD linear systems*, in Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium, 2011, pp. 590–598.

[20] M. G. Larson and A. Målqvist, *Adaptive variational multiscale methods based on a posteriori error estimation: Energy norm estimates for elliptic problems*, Comp. Methods Appl. Mech. Engrg., 196 (2007), pp. 2313–2324.

[21] O. E. Livne and A. Brandt, *Lean algebraic multigrid (LAMG): Fast graph laplacian linear solver*, SIAM J. Sci. Comput., 34 (2012), pp. B499–B522.

[22] A. Målqvist and D. Petersheim, *Localization of elliptic multiscale problems*, Math. Comp., 83 (2014), pp. 2583–2603.

[23] B. Moghaddam, Y. Weiss, and S. Avidan, *Spectral bounds for sparse PCA: Exact and greedy algorithms*, Adv. Neural Inf. Process. Syst., 18 (2006), p. 915.

[24] A. Myronenko and X. Song, *Point set registration: Coherent point drift*, IEEE Trans. Pattern Anal. Mach. Intell., 32 (2010), pp. 2262–2275.

[25] M. E. Newman, *Modularity and community structure in networks*, Proc. Natl. Acad. Sci., 103 (2006), pp. 8577–8582.

[26] A. Y. Ng, M. I. Jordan, Y. Weiss, et al., *On spectral clustering: Analysis and an algorithm*, NIPS, 14 (2001), pp. 849–856.

[27] H. Owaidi, *Multigrid with rough coefficients and multiresolution operator decomposition from hierarchical information games*, SIAM Rev., 59 (2017), pp. 99–149.

[28] H. Owaidi and C. Scovel, *Universal Scalable Robust Solvers from Computational Information Games and Fast Eigenspace Adapted Multiresolution Analysis*, preprint, arXiv:1703.10761, 2017.

[29] Y. Saad, *Numerical Methods for Large Eigenvalue Problems: Revised Edition*, SIAM, Philadelphia, PA, 2011.

[30] R. S. Sampath and G. Biros, *A parallel geometric multigrid method for finite elements on octree meshes*, SIAM J. Sci. Comput., 32 (2010), pp. 1361–1392.

[31] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, MIT Press, Cambridge, MA, 2002.

[32] D. A. Spielman and S.-H. Teng, *Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems*, in Proceedings of the Thirty-Sixth Annual ACM Symposium on Theory of Computing, ACM, 2004, pp. 81–90.

[33] D. A. Spielman and S.-H. Teng, *A Local Clustering Algorithm for Massive Graphs and Its Application to Nearly-Linear Time Graph Partitioning*, preprint, arXiv:0809.3232, 2008.

[34] D. A. Spielman and S.-H. Teng, *Spectral sparsification of graphs*, SIAM J. Comput., 40 (2011), pp. 981–1025.

[35] D. A. Spielman and S.-H. Teng, *A local clustering algorithm for massive graphs and its application to nearly linear time graph partitioning*, SIAM J. Comput., 42 (2013), pp. 1–26.
[36] D. A. Spielman and S.-H. Teng, Nearly linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems, SIAM J. Matrix Anal. Appl., 35 (2014), pp. 835–885.

[37] K. Stüben, A review of algebraic multigrid, J. Comput. Appl. Math., 128 (2001), pp. 281–309.

[38] U. Trottenberg, C. W. Oosterlee, and A. Schuller, Multigrid, Academic Press, New York, 2000.

[39] P. Vanek, J. Mandel, and M. Brezina, Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems, Computing, 56 (1996), pp. 179–196.

[40] P. Wesseling and C. W. Oosterlee, Geometric multigrid with applications to computational fluid dynamics, J. Comput. Appl. Math., 128 (2001), pp. 311–334.

[41] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin, Graph embedding and extensions: A general framework for dimensionality reduction, IEEE Trans. Pattern Anal. Mach. Intell., 29, 2007.

[42] H. Zou, T. Hastie, and R. Tibshirani, Sparse principal component analysis, J. Comput. Graph. Statist., 15 (2006), pp. 265–286.