# Positive Definite Matrices: Compression, Decomposition, Eigensolver, and Concentration

Thesis by
De Huang

In Partial Fulfillment of the Requirements for the
Degree of
Doctor of Philosophy


CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2020
Defended May 14, 2020

De Huang
ORCID: 0000-0003-4023-9895

# ACKNOWLEDGEMENTS

First of all, I would like to express my deepest gratitude to my adviser, Prof. Thomas Yizhao Hou, for his consistently wholehearted mentoring, supporting and inspiring. Prof. Hou introduced me to the frontier of many important fields including multiscale PDE, fluid dynamics, data analysis and machine learning, and I have truly enjoyed my interdisciplinary research among these fascinating areas. Prof Hou's enthusiastic dedication to mathematics and steadfast resolution in solving hard problems have aroused my passion for research, toughened my will in hard-working and moreover, taught me how to become a better man. Beyond the academic, Prof. Hou is not only a good mentor but also a caring friend, continuously providing me a lot of helps in my personal life. To this point, I would also like to thank Prof. Hou's wife, Prof. Chang, for sharing with us her knowledge in Buddhism and for bringing us delicious food on occasion.

I want to thank Prof. Houman Owhadi for he has been an inspiration in a large portion of the work in my thesis. He has taught me with patience many useful techniques in multiscale analysis and probability theory.

I must also thank Prof. Joel Tropp, who guided me to the realm of random-matrix theories. His excellent skills and keen insight in matrix analysis have greatly helped me build up my understanding in this field and develop my own works.

I am grateful to Prof. Andrew Stuart and Prof. Peter Schroeder for kindly serving in my thesis committee. They have provided me with invaluable advice on my thesis.

I feel honored to have had stimulating discussions with many brilliant mathematicians, especially Prof. Oscar Bruno, Prof. Venkat Chandrasekaran, Prof. Thomas Vidick, Prof. Stanley Osher, Prof. Alexander Kiselev, Prof. Vladimír Šverák, Prof. Jack Xin, Prof. Jonathan Niles-Weed, Prof. Rachel Ward, Dr. Kachun Lam, Dr. Pengchuan Zhang, Dr. Pengfei Liu, Mr. Yifan Chen, Mr. Jiajie Chen, Mr. Shumao Zhang, Ms. Ziyun Zhang, and more. Their knowledge truly inspired me. Besides that, I want to thank all the professors in the wonderful department of Computational and Mathematical Science at Caltech who have never shown hesitation in answering my questions.

Thanks are also due to all staff members of the CMS department, especially Maria Lopez, Carmen Sirois, Diana Bohler and Sydney Garstang. They have been taking good care of me in various ways in the five three years.

Coming to friends, I have been truly fortunate to have their company during my days of joys and days of bitterness. First of all, there is Muzhe Zeng, my closest confidant and best friend, with whom I share the happiest memories of more than ten years. He is like the mainspring to my gears of joy and the piston to my engine of passion.

Then there are my three closest friends at Caltech, Jialin Song, Florian Schaefer and Mandy Huo. We have had a lot of unforgettably interesting discussions on a variety of topics, from the deepest mysteries of humanity to the farthest void of the universe. Moreover, they treated me like true family.

Besides, there is a long list of lovely names including Yufei Ou, Qingcan Wang, Zichao Long, Yingying Hu, Jinyang Huang, Zhuoheng Liu, Jiahui Tang, Qingwen Xu, Andrea Coladangelo, Navid Azizan, Xinying Ren, Karena Cai, Bo Sun, Xinliang Liu, and many many more.

Last but not least, I owe tons of thanks to my fairylike girlfriend, Su Wang. You have been supporting me all along and bringing me endless happiness. I could not have survived the last year of my strenuous Ph.D. journey without you.

This thesis is especially dedicated to my mother, Ms. Youping Zuo, the strongest woman I have ever known. I am truly grateful to her for bringing me up the way she did. If there is any goodness in my character, it is from her. She gives me all her support to my pursuit of knowledge, even though it means I have to leave her in years of worry and missing. I love you, Mom.

# ABSTRACT

For many decades, the study of positive definite (PD) matrices has been one of the most popular subjects among a wide range of scientific researches. A huge mass of successful models on PD matrices have been proposed and developed in the fields of mathematics, physics, biology, etc., leading to a celebrated richness of theories and algorithms. In this thesis, we draw our attention to a general class of PD matrices featured by the generic form $A = \sum_{k=1}^{m} E_k$, where $\{E_k\}_{k=1}^{m}$ is a finite sequence of positive semidefinite matrices reflecting the topological and structural nature of $A$. For this class of PD matrices, we will develop theories and algorithms on operator compression, multilevel decomposition, eigenpair computation, and spectrum concentration. We divide these contents into three main parts.

In the first part, we propose an adaptive fast solver for the preceding class of PD matrices which includes the well-known graph Laplacians. We achieve this by establishing an adaptive operator compression scheme and a multiresolution matrix factorization algorithm which have nearly optimal performance on both complexity and well-posedness. To develop our methods, we introduce a novel notion of energy decomposition for PD matrices and two important local measurement quantities, which provide theoretical guarantee and computational guidance for the construction of an appropriate partition and a nested adaptive basis.

In the second part, we propose a new iterative method to hierarchically compute a relatively large number of leftmost eigenpairs of a sparse PD matrix under the multiresolution matrix compression framework. We exploit the well-conditioned property of every decomposition components by integrating the multiresolution framework into the Implicitly Restarted Lanczos method. We achieve this combination by proposing an extension-refinement iterative scheme, in which the intrinsic idea is to decompose the target spectrum into several segments such that the corresponding eigenproblem in each segment is well-conditioned.

In the third part, we derive concentration inequalities on partial sums of eigenvalues of random PD matrices by introducing the notion of $k$-trace. For this purpose, we establish a generalized Lieb's concavity theorem, which extends the original Lieb's concavity theorem from the normal trace to $k$-traces. Our argument employs a variety of matrix techniques and concepts, including exterior algebra, mixed discriminant, and operator interpolation.

# PUBLISHED CONTENT AND CONTRIBUTIONS

[1] Thomas Y. Hou, De Huang, Ka Chun Lam, and PengChuan Zhang. An adaptive fast solver for a general class of positive definite matrices via energy decomposition. *Multiscale Modeling & Simulation*, 16(2):615–678, 2018. URL https://doi.org/10.1137/17M1140686.
D.H. proposed this project, generalized the operator compression method and the multiresolution matrix decomposition framework, designed the adaptive partitioning algorithm, proved the main theoretical results, conducted the numerical experiments, and participated in the writing of the manuscript.

[2] Thomas Y. Hou, De Huang, Ka Chun Lam, and Ziyun Zhang. A fast hierarchically preconditioned eigensolver based on multiresolution matrix decomposition. *Multiscale Modeling & Simulation*, 17(1):260–306, 2019. URL https://doi.org/10.1137/18M1180827.
D.H. proposed this project, designed the hierarchical framework of the eigensolver, proposed the spectral preconditioner, proved the main theoretical results, conducted the numerical experiments, and participated in the writing of the manuscript.

[3] De Huang. A generalized Lieb's theorem and its applications to spectrum estimates for a sum of random matrices. *Linear Algebra and its Applications*, 579:419 – 448, 2019. ISSN 0024-3795. URL https://doi.org/10.1016/j.laa.2019.06.013.
D.H. is the sole author of the manuscript.

[4] De Huang. Generalizing Lieb's concavity theorem via operator interpolation. *Advances in Mathematics*, 369:107208, 2020. ISSN 0001-8708. URL https://doi.org/10.1016/j.aim.2020.107208.
D.H. is the sole author of the manuscript.

# TABLE OF CONTENTS

# LIST OF ILLUSTRATIONS

# LIST OF TABLES

*Chapter 1*

# INTRODUCTION

## 1.1 Operator Compression and Fast Linear Solver

Fast algorithms for solving symmetric positive definite (PD) linear systems have found broad applications across both theories and practice, including machine learning [11, 21, 106], computer vision [10, 15, 135], image processing [5, 23, 81], computational biology [32, 83], etc. For instance, solving linear systems of graph Laplacians, which has a deep connection to the spectral properties of the underlying graphs, is one of the foundational problems in data analysis. Performing finite element simulation on a wide range of physical systems will introduce the corresponding stiffness matrix, which is also symmetric and positive definite.

Many related works have drawn inspiration from the spectral graph theory [30, 84] in which the spectrum and the geometry of graphs are found to be highly correlated. By computing the spectrum of the graph, the intrinsic geometric information can be directly obtained and various applications can be found [16, 57]. However, the price for finding the spectrum of graph is relatively expensive as it involves solving global eigen problems. On the contrary, the Algebraic Multigrid method [118, 123, 125] 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 the first part of the thesis, we propose an adaptive fast solver for a general class of PD 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 a PD 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 guidances 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, PD linear systems. In particular, following the work in [88], we propose a nearly-linear time fast solver for general PD matrices. Given the prescribed well-posedness requirement (i.e., the **condition factor**), every component from MMD will be a well-conditioned, lower dimensional PD linear system. Any generic iterative solver can then be applied in parallel to obtain the approximated solution of the given arbitrary PD matrix satisfying the prescribed accuracy.

**Problem Setting**

Given an $n \times n$ PD matrix $A$, our ultimate goal is to develop a fast algorithm to efficiently solve $Ax = b$, or equivalently, compress the solver $A^{-1}$ with a desired compression error. We make the following assumptions on the matrix $A$. First of all, $\lambda_{\min}(A) = O(1)$ 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 (PSD) matrices, i.e. $A = \sum_{k=1}^{m} E_k$ for a sequence $\{E_k\}_{k=1}^{m}$ of PSD 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, such sparsity can be described by the requirement

$$\#\mathcal{N}_k(i) = O(k^d), \ \forall i,$$

where $\#\mathcal{N}_k(i)$ is the number of vertices near the vertex $i$ with logic distance no greater than $k$ 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 PSD matrices and can be obtained automatically if $G$ is given.

Another illustrative example is the patch-wise stiffness matrix of a finite element from the discretization of partial differential equation (PDE) using finite element type methods.

**Operator Compression**

To compress the operator $A^{-1}$ (where $A$ satisfies the above assumptions) with a desired error bound, we adopt the idea of constructing a modified coarse space as proposed in [53, 75, 88]. The procedure is summarized in Figure 1.1. 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 requisite 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 $\mathcal{P} = \{P_j\}_{j=1}^{M}$ (e.g., graph partitioning in the case of graph Laplacian) which reveals the structural property of $A$.

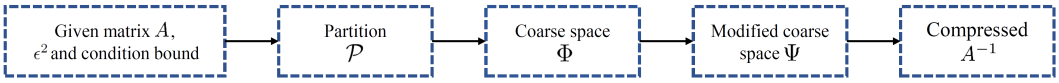



Figure 1.1: Process flowchart for compressing $A^{-1}$.

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 [53, 88], having such requirement, together with the modification of coarse space $\Phi$ into $\Psi = A^{-1}(\Phi)$, we have $\|A^{-1} - P_\Psi^A A^{-1}\|_2 \leq \epsilon^2$. This affirms us 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 $\|x - P_{\Phi_j} x\|_2$, which in turns gives a local computation procedure for checking the qualification of the local $\Phi_j$. Specifically, we introduce the **error factor**

$$\varepsilon(\mathcal{P}, q) = \max_{P_j \in \mathcal{P}} \frac{1}{\sqrt{(\lambda_{q+1}(P_j))}}$$

(where $\lambda_{q+1}(P_j)$ corresponds to the $(q+1)^{\text{th}}$ smallest eigenvalue of some eigen problem 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 $P_j \in \mathcal{P}$. 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 eigen problem. 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 $A_{\text{st}}$ satisfies $\kappa(A_{\text{st}}) \leq \max_{P_j \in \mathcal{P}} \delta(P_j)\|A^{-1}\|_2$.

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 significantly reduced.

**Multiresolution Matrix Decomposition**

Having a generic operator compression scheme, we follow the idea in [88] to extend the compression scheme hierarchically to form a multiresolution matrix decomposition (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 PD matrices with time

complexity

$$O(nnz(A) \cdot \log n \cdot \left(\log \epsilon^{-1} + \log n\right)^{c} \log \epsilon^{-1}),$$

where $nnz(A)$ is the number of nonzero entries in $A$ 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. Our study shows that 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 a purpose similar to that of 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 would like to emphasize that with an appropriate choice of partitioning, the sparsity and the well-posedness properties of the compressed operator can be inherited through layers. This nice property enables us to decompose the original problem of solving $A^{-1}$ into sets of independent problems with similar complexities and condition numbers, which favors the parallel implementation of the solver.

**Related Previous Works**

In the past few years, several works relevant to the compression of elliptic operators with heterogeneous and highly varying coefficients have been proposed. Målqvist and Peterseim et al. [63, 75] constructed localized multiscale basis functions from the modified coarse space $V_H^{ms} = V_H - \mathfrak{F}V_H$, where $V_H$ is the original coarse space spanned by conforming nodal basis, and $\mathfrak{F}$ is the energy projection onto the space $(V_H)^{\perp}$. The exponential decaying property of this modified basis has also been shown both theoretically and numerically. Meanwhile, a beautifully insightful work by Owhadi [88] reformulated 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 a 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 [75]. In addition, the basis of $\Phi$ is generalized to nonconforming measurement functions and the gamblets still enjoy the exponential decay property which makes localization possible. Hou and Zhang in [53] extended these works such that localized basis functions can also be constructed for higher-order strongly elliptic operators. Owhadi further generalized these frameworks to a more unified methodology for arbitrary elliptic operators on Sobolev spaces in [89] using the Gaussian process interpretation. Note that for the

above-mentioned works, since the problems they considered are 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 PD 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. Recently, Schäfer et al. [104] proposed a near-linear running time algorithm to compress a large class of dense kernel matrices $\Theta \in \mathbb{R}^{n\times n}$. The authors also provided rigorous complexity analyses and showed that the storage complexity of the proposed algorithm is $O(n\log(n)\log^d(n/\epsilon))$ and the running time complexity is $O(n\log^2(n)\log^{2d}(n/\epsilon))$, where $d$ is the intrinsic dimension of the problem.

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 widely used Geometric Multigrid (GMG) methods [28, 103, 129] 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 made. Numerous methods ranging from geometry specific to purely algebraic/ matrix-based approach have been developed (See [7, 44, 118] for review). Using the tools of compressing the operator $A^{-1}$ possessing multiple scale features, Owhadi in [88] also proposed 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 was proposed. This in turn provides a near-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 [89], Owhadi and Scovel extended the result to problems with general PD 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 $\Phi$ still presents some essential difficulty 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 level-wisely 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 Algebraic Multigrid (AMG) methods [118, 123, 125] take advantage of the connectivity information given in the matrix $A$ itself to define intergrid transfer operators, which avoids the direct construction of the restriction and relaxation operators in GMG methods. Intuitively, the connectivity information reveals 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 [71], 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-typed 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 PD 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 PD linear systems.

Another inspiring stream of nearly-linear time algorithm for solving graph Laplacian system was given by Spielman and Teng [112–115]. With the innovative discoveries in spectral graph theory and graph algorithms, such as the fast construction of low-stretch spanning trees and clustering scheme, they successfully employed all these important techniques in developing an effective preconditioned iterative solver. Later, Koutis, Miller and Peng [59] followed these ideas and simplified the solver with computation complexity $O(m \log n \log\log n \log \epsilon^{-1})$, 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 PD matrix (i.e., the graph Laplacian in this case) hierarchically with the control of sparsity and well-posedness.

## 1.2 Hierarchically Preconditioned Eigensolver

The computation of eigenpairs for large and sparse matrices, particularly for positive semidefinite matrices (PSD) matrices, is one of the most fundamental tasks in many scientific applications. For example, the leftmost eigenpairs (i.e., the eigenpairs associated with the $N$ smallest eigenvalues for some $N \in \mathbb{N}$) of a graph Laplacian $L$ help reveal the topological information of the corresponding network from real data. One illustrative example is that the multiplicity of the smallest eigenvalue $\lambda_1$ of $L$ coincides with the number of the connected components of the corresponding graph $G$. In particular, the second-smallest eigenvalue of $L$ is well known as the algebraic connectivity or the Fiedler value of the graph $G$, which is applied to develop algorithms for graph partitioning [30, 82, 84]. Another important example regarding the use of leftmost eigenpairs is the computation of betweenness centrality of graphs as mentioned in [12, 17, 18]. Computing the leftmost eigenpairs of large and sparse PSD matrices also stems from the problem of predicting electronic properties in complex structural systems [41]. Such prediction is achieved by solving the Schrödinger equation $\mathcal{H}\Psi = \mathcal{E}\Psi$, where $\mathcal{H}$ is the Hamiltonian operator for the system, $\mathcal{E}$ corresponds to the total energy and $|\Psi(r)|^2$ represents the charge density at location $r$. Solving this equation using the self-consistent field requires computing the eigenpairs of $\mathcal{H}$ repeatedly, which dominates the total computation cost of the overall iterations. Thus, an efficient algorithm to solve the eigenproblem is indispensable. Usage of leftmost eigenpairs can also be found in vibrational analysis in mechanical engineering [78]. In [31], authors also suggest that the leftmost eigenpairs of the covariance matrix between residues are important to extract functional and structural information about protein families. Efficient algorithms for computing $p$ smallest eigenpairs for relatively large $p$ are therefore crucial in various applications.

### Iterative Methods

As most of the linear systems from engineering problems or networks are typically large and sparse in nature, iterative methods are preferred. Recently, several efficient algorithms have been developed to obtain leftmost eigenpairs of $A$. These include the Jacobi–Davidson method [108], implicit restarted Arnoldi/Lanczos method [22, 65, 110], and the deflation-accelerated Newton method [13]. All these methods give promising results [12, 76], especially for finding a small number of leftmost eigenpairs. Other methods for computing accurate leftmost eigenpairs using hierarchical refinement/correction techniques were proposed in [69, 133, 134].

However, as reported in [76], the Implicit Restarted Lanczos Method (IRLM) is still the best performing algorithm when a large number of smallest eigenpairs are required. Therefore, it is highly desirable to develop a new algorithm, based on the architecture of the IRLM, that can further optimize the performance.

The main purpose of this part of work is to explore the possibility of exploiting the advantageous energy decomposition framework under the architecture of the IRLM. In particular, we propose a new spectrum-preserving preconditioned hierarchical eigensolver for computing a large number of the leftmost eigenpairs. This eigensolver takes full advantage of the intrinsic structure of the given matrix, the nice spectral property in the Lanczos procedure and also the preconditioning characteristics of the Conjugate Gradient (CG) method. Given a sparse symmetric positive matrix $A$ which is assumed to be energy decomposable (See Section 2.1 for details), we integrate the well-behaved matrix properties that are inherited from the MMD with IRLM. The preconditioner we propose for the CG method (which hence becomes the Preconditioned Conjugate Gradient (PCG) method) can also preserve the narrowed residual spectrum of $A$ during the Lanzcos procedure. Throughout this thesis, theoretical performance of our proposed algorithm is analyzed rigorously and we conduct a number of numerical experiments to verify the efficiency and effectiveness of the algorithm in practice. To summarize, our contributions are three folds:

- We establish a hierarchical framework to compute a relatively large number of leftmost eigenpairs of a sparse symmetric positive matrix. This framework employs the MMD algorithm to further optimize the performance of IRLM. In particular, a specially designed spectrum-preserving preconditioner is introduced for the PCG method to compute $x = A^{-1}b$ for some vector $b$.

- The proposed framework improves the running time of finding the $m_{tar}$ leftmost eigenpairs of a matrix $A \in \mathbb{R}^{n\times n}$ from $O(m_{tar}\cdot\kappa(A)\cdot nnz(A)\log\frac{1}{\varepsilon})$ (which is achieved by the classical IRLM) to $O\left(m_{tar}\cdot nnz(A)\cdot(\log\frac{1}{\varepsilon}+\log n)^C\right)$, where $\kappa(A)$ is the condition number of $A$, $nnz(\cdot)$ is the number of nonzero entries and $C$ is some small constant independent of $m_{tar}, nnz(A)$ and $\kappa(A)$.

- We also provide a rigorous analysis on both the accuracy and the asymptotic computational complexity of our proposed algorithm. This ensures the correctness and efficiency of the algorithm even in large-scale, ill-conditioned scenarios.

In many practical applications, the operator $A$ may not be explicitly stored entry-wisely and only the evaluation of $Ax$ is available. In this situation, our proposed algorithm also works as it only requires the storage of the stiffness matrices corresponding to some hierarchical basis $\Psi$ according to the accuracy requirement. To construct these stiffness matrices, we only need to evaluate $Ax$. Therefore, for the ease of discussion, we simply assume that the given operator $A$ is a finite-dimensional accessible matrix.

**Overview of Our Method**

In this part of work, we propose and develop an iterative scheme for computing a relatively large number of the left most eigenpairs of a PSD matrix, under the framework of operator compression and decomposition introduced in the first part of the thesis. Note that we can transfer a PSD matrix into a PD matrix by adding to it a constant multiple of the identity matrix, which does not change the eigenvectors and only shifts all eigenvalues by one uniform constant. Under our preceding framework, we can decompose the inverse $A^{-1}$ of a PD matrix $A \in \mathbb{R}^{n\times n}$ into

$$A^{-1} = P^A_{\mathcal{U}} A^{-1} + P^A_{\Psi} A^{-1} := P^A_{\mathcal{U}} A^{-1} + \Theta,$$

where $[\mathcal{U}, \Psi]$ corresponds to a basis of $\mathbb{R}^n$; $P^A_{\mathcal{U}}$ and $P^A_{\Psi}$ are the corresponding subspace projections. Recursively, we can also consider $\Theta$ as a "new" $A^{-1}$ and further decompose $\Theta$ in the same manner. This will give a MMD of $A^{-1} = \sum_{k=1}^{K} P^A_{\mathcal{U}^{(k)}} A^{-1} + \Theta^{(K)}$. To illustrate, we first consider a 1-level decomposition, i.e., $K = 1$. One important observation regarding this decomposition is that the spectrum of the original operator $A^{-1}$ resembles that of the compressed operator $\Theta$. In particular, if $\lambda_{i,\Theta}$ is the $i^{\text{th}}$ smallest eigenvalue of $\Theta$ and $\zeta_{i,\Theta}$ is the corresponding eigenvector, then $(\lambda^{-1}_{i,\Theta}, \zeta_{i,\Theta})$ is a good approximation of $(\lambda_i^{-1}, q_i)$ for small $\lambda_i$, where $(\lambda_i^{-1}, q_i)$ denotes the $i^{\text{th}}$ eigenpair of $A^{-1}$. These approximate eigenpairs $(\lambda^{-1}_{i,\Theta}, \zeta_{i,\Theta})$ can then be used as the initial approximation of the targeted eigenpairs. Notice that compression errors are introduced into these eigenpairs by the matrix decomposition. Therefore, a refinement procedure should be carried out to reduce these errors up to the prescribed accuracy. Once we obtain the refined eigenpairs, we may extend the spectrum in order to obtain the targeted number of eigenpairs. As observed in [76], the IRLM is the best performing algorithm when a large number of eigenpairs are considered, we therefore employ the Krylov subspace extension technique to extend spectrum up to some prescribed control of the well-posedness. Intuitively, the MMD decomposes the spectrum of $A^{-1}$ into different segments of different scales. Using a

subset of the decomposed components to approximate $A^{-1}$ yields a great reduction of the relative condition number. Thus, we can further trim down the complexity of the IRLM by approximating $A^{-1}$ during the shifting process.

To generalize from the 1-level algorithm to the $K$-level algorithm, we develop a hierarchical scheme to compute the leftmost eigenpairs of an energy decomposable matrix. Given the $K$-level multiresolution decomposition $\{\Theta^{(k)}\}_{k=1}^{K}$ of an energy decomposable matrix $A$, we first compute the eigen decomposition $[V_{ex}^{(K)}, D_{ex}^{(K)}]$ of $\Theta^{(K)}$ (with dimension $N^{(K)}$) corresponding to the coarsest level by using some standard direct method. Then we propose a compatible refinement scheme for both $V_{ex}^{(K)}$ and $D_{ex}^{(K)}$ to obtain $V_{ini}^{(K-1)}$ and $D_{ini}^{(K-1)}$, which will then be used as the initial spectrum in the consecutive finer level. The efficiency of the cross-level refinement is achieved by a modified version of the orthogonal iteration with the Ritz Acceleration, where we exploit the proximity of the eigenspace across levels to accelerate the CG/PCG method within the refinement step. Using this refined initial spectrum, our second stage is to extend spectrum up to some prescribed control of the well-posedness using the Implicit Restarted Lanczos architecture. Recall that a shifting approach is introduced to reduce the iteration number for the extension, which again requires solving $A^{(K-1)}x = w$ with the PCG method in each iteration. However, the preconditioner for PCG when we are solving for $A^{(K-1)}w$ must be chosen carefully. Otherwise the orthogonal property brought about by the Krylov subspace methods may not be utilized and a large CG iteration number will be required (See Section 4.7). In view of this, we propose a spectrum-preserving hierarchical preconditioner $M^{(K-1)} := (\Psi^{(K-1)})^T\Psi^{(K-1)}$ for accelerating the CG iteration during the Lanczos iteration. In particular, we can show that using the preconditioner $M^{(K-1)}$, the number of PCG iterations to achieve a relative $\varepsilon$ in $A^{(K-1)}$-norm can be controlled in terms of the condition factor $\delta(\mathcal{P})$ (from the energy decomposition of the matrix) and an extension threshold $\mu_{ex}^{(K-1)}$.

This process then repeats hierarchically until we reach the finest level. Under this framework, the condition numbers of all the corresponding operators within each level are controlled. The overall accuracy of our proposed algorithm is also determined by the prescribed compression error at the highest level.

**Related Previous Works**

Several important iterative methods have been proposed to tackle the eigenproblems of PD matrices. One of the well-established algorithms is the Implicitly Restarted

Lanczos Method (IRLM) (or the Implicitly Restarted Arnoldi Method (IRAM) for asymmetrical sparse matrices), which has been implemented in various popular scientific computing packages like MATLAB, R and ARPACK. The IRLM combines both the techniques of the implicitly shifted QR method and the shifting of the operators to avoid the difficulties for obtaining the leftmost eigenpairs. Another popular algorithm for finding leftmost eigenpairs is the Jacobi–Davidson method. The main idea is to minimize the Rayleigh quotient $q(x) = \frac{x^T Ax}{x^T x}$ using a Newton-type methodology. Efficacy and stability of the algorithm are then achieved by using a projected simplification of the Hessian of the Rayleigh quotient namely, $\tilde{J}(x_k) := (I - x_k x_k^T)(A - q(x_k)I)(I - x_k x_k^T)$ with the update of $x_k$ given by

$$x_{k+1} = x_k - \tilde{J}(x_k)^{-1}(Ax_k - q(x_k)x_k). \tag{1.1}$$

Notice that the advantage of such approach is the low accuracy requirement for solving (1.1). A parallel implementation of this algorithm was also proposed [99]. In [13], the authors proposed the Deflation-Accelerated Conjugate Gradient (DACG) method designed for solving the eigenproblem of PD matrices. The main idea is to replace the Newton's minimization procedure of the Rayleigh quotient $r(x)$ by the nonlinear CG method which avoids solving linear systems within the algorithm. A comprehensive numerical comparison between the three algorithms was reported in [12]. Recently, Martínez [76] studied a class of tuned preconditioners for accelerating both the DACG and the IRLM for the computation of the smallest set of eigenpairs of large and sparse PD matrices. However, as reported in [76], the IRLM still outperforms the others when a relatively large number of leftmost eigenpairs is desired. By virtue of this, we are motivated to develop a more efficient algorithm particularly designed for computing a large number of leftmost eigenpairs.

Another class of methods is designed for constructing localized/compressed eigenmodes that capture the space spanned by the true leftmost eigenvectors. One of the representative pioneer works was proposed by Ozoliņš et al. in [90]. The goal of this work is to obtain a spatially localized solution of a class of problems in mathematical physics by constructing the compressed modes. In particular, finding these localized modes can be formulated as an optimization problem

$$\Psi_N = \arg\min_{\hat{\Psi}^N} \frac{1}{\mu}\|\Psi_N\|_1 + \mathrm{Tr}(\hat{\Psi}_N^T H \hat{\Psi}_N) \quad \text{such that} \quad \hat{\Psi}_N^T \hat{\Psi}_N = I.$$

The authors of [90] proposed an algorithm based on the split Bregman iteration to solve the $L_1$ minimization problem. Replacing the Hamiltonian operator $H$ by a

graph Laplacian matrix, one obtains the $L_1$ regularized Principal Component Analysis. In particular, if there is no $L_1$ regularization term in the optimization problem, the optimal $\Psi_N$ will be the first $N$ eigenvectors of $A$. Furthermore, when $N$ is reasonably larger than some integer $m_{tar}$, the localized basis functions $\Psi_N$ produced with $L_1$ regularization can approximately span the $m_{tar}$ leftmost eigenspace (i.e., the eigenspace spanned by the $m_{tar}$ leftmost eigenvectors). Similarly, the MMD framework provides us the hierarchical and sparse/localized basis $\Psi$. These localized basis functions capture the compressed modes and eventually provide us a convenient way to control the complexity of the eigensolver.

### 1.3 Concentration of Eigenvalue Sums

As we have discussed in the preceding sections, theories and algorithms on the leftmost eigenpairs of PSD matrices, have drawn great attention from a wide range of studies, due to their importance in many scientific modelings. In the study of physical Hamiltonian systems, the leftmost eigenpairs of a Hamiltonian operator represent the ground states and their energies [61, 128]. Computing the leftmost eigenpairs is therefore a fundamental task for understanding a Hamiltonian system. As we have mentioned in the previous section, the smallest eigenvalues of a graph Laplacian reflect the connectivity of the graph, and the corresponding eigenvectors divide the graph into closely connected clusters. Based on this, a great numbers of spectral clustering techniques have been developed over the last decades [29, 84, 127].

The importance of developing efficient algorithms for computing the leftmost eigenpairs of PSD matrices is then beyond question. However, the efficiency of the hierarchical eigensolver we proposed, and of many other iterative methods, relies heavily on the sparsity of the target PSD matrix. When we deal with a large and dense matrix in practice, random sparsification or random sampling provides an effective way to remarkably reduce the computational cost [111, 113]. The purpose of random sparsification is to reduce the complexity of a system without significantly perturbing its spectrum. Then by applying eigensolvers to the sparsified matrices, we can still be confident that the resulting eigenpairs are close to the ground truth. The practicability of these random approaches is guaranteed by the well-established theories of matrix concentration.

In this thesis, we will establish concentration results on the sum of the smallest eigenvalues of PSD matrices. The reason why we study this quantity is that it carries meaningful algebraic and geometric information of the matrix. By an extended

version of the Courant-Fisher characterization of eigenvalues, the sum of the $k$ smallest eigenvalues of a PSD matrix $A$ (denoted by $S_k(A)$) is the minimum of the optimization problem

$$\min_{Q\in\mathbb{C}^{n\times k}, Q^*Q=I_k} \mathrm{Tr}[Q^*AQ],$$

and this minimum is achieved by the orthogonal matrix $V = [v_1, v_2, \cdots, v_k] \in \mathbb{C}^{n\times k}$ whose columns are the k leftmost eigenvectors of $A$. Therefore, if we obtain an approximation $\hat{V}$ of $V$, the accuracy of this approximation can be measured by the ratio $\mathrm{Tr}[\hat{V}^*A\hat{V}]/S_k(A)$, which requires knowing the value of $S_k(A)$. When $A$ is large and dense, computing $S_k(A)$ can be a computationally intractable problem. However, if we can efficiently compute the smallest eigenvalues of a random sparsification $\tilde{A}$ of $A$ and if we have concentration guarantee on $S_k(\tilde{A})$ such that $S_k(\tilde{A}) - S_k(A)$ is small, then we can use $S_k(\tilde{A})$ as a substitute to measure the accuracy of approximated eigenvectors.

Another illustration of the importance of the sum of the smallest eigenvalues arises in the connectivity study of graphs. By spectral theory, the number of zero eigenvalues of the Laplacian of a graph indicates the number of connected components in the graph. A relaxed version is that the number of the Laplacian's eigenvalues that are close to zero indicates the number of major clusters in the graph. Based on this observation, many researchers have developed clustering methods by investigating the smallest eigenvalues of graph Laplacians. Therefore, if the sum of the smallest eigenvalues has a concentration property, we can indirectly study the clustering of a graph by only looking at the Laplacian of an associated sparsified graph. Moreover, in many cases of interest the number $k$ of clusters is assumed to be known a priori [29, 96, 98], then the smallness of the sum of the $k$ smallest of a sparsified Laplacian is a good indicator that it preserves the connectivity property of the original graph.

We remark that the rightmost eigenpairs of PSD matrices also play a significant role in many applications. In fact, studying the outstanding rightmost eigenpairs of a correlation matrix is the foundation of the Principal Component Analysis [55, 56], which aims to extract an intrinsic low-dimensional structure from high-dimensional data. We will therefore simultaneously develop concentration results on the sum of the smallest eigenvalues and the sum of the largest eigenvalues.

**Matrix Concentration Inequalities**

Matrix concentration describes how likely a random matrix is close to its expectation. As noncommutative extensions of their scalar counterparts, matrix concentration

inequalities have been widely studied through many efforts and have had a profound impact on a wide range of areas in computational mathematics and statistics. In particular, many important results on concentration phenomena of sums of independent random matrices [73, 87, 101, 121] and matrix-valued martingale sequences [86, 94, 120] have been developed over the past decades; we refer to the monograph [122] by Tropp for a detailed overview on this subject and an extensive bibliography.

These matrix concentration results provide rich theoretical supports for studies and developments in stochastic models and algorithms for random matrices [77, 130] in fields ranging from quantum physics [43] to financial statistics [62, 95]. A typical example is the study of clustering of random graphs [1, 38] arising from research on social networks [79], image classification [9, 20] and so on.

In our setting, we are interested in the concentration phenomena of a class of random PSD matrices that admit an energy decomposition. To be specific, we want to study how far does the spectrum of a random PSD matrix $Y$ deviate from the spectrum of its expectation $\mathbb{E}Y$, where $Y = \sum_{i=1}^{m} X^{(i)}$ is the sum of a sequence of independent random PSD matrices $\{X^{(i)}\}_{i=1}^{m}$. For this type of random matrices, concentration inequalities on the extreme eigenvalues have been well developed [122]. For example, the following Chernoff-type tail bounds regarding the largest and the smallest eigenvalues are due to Tropp [121, Theorem 1.1]:

$$\mathbb{P}\{\lambda_{\max}(Y) \geq (1+\varepsilon)\lambda_{\max}(\mathbb{E}Y)\} \leq n\left(\frac{e^{\varepsilon}}{(1+\varepsilon)^{1+\varepsilon}}\right)^{\lambda_{\max}(\mathbb{E}Y)/c}, \quad \varepsilon \geq 0,$$

$$\mathbb{P}\{\lambda_{\min}(Y) \leq (1-\varepsilon)\lambda_{\min}(\mathbb{E}Y)\} \leq n\left(\frac{e^{-\varepsilon}}{(1-\varepsilon)^{1-\varepsilon}}\right)^{\lambda_{\min}(\mathbb{E}Y)/c}, \quad \varepsilon \in [0,1),$$

where $n$ is the dimension of $Y$ and $c = \max_i \|X^{(i)}\|$. This inequality indicates that, with a high probability, the largest (or smallest) eigenvalue of $Y$ still provides a good approximation to that of $\mathbb{E}Y$.

The main purpose of this part of the thesis is to generalize these concentration inequalities on the extreme eigenvalues to their counterparts on the sum of the $k$ largest (or smallest) eigenvalues for any natural number $k$. In particular, we will extend the preceding Tropp's inequalities to the following new Chernoff-type tail

bounds:

$$\mathbb{P}\left\{\sum_{i=1}^{k}\lambda_i(Y)\ \geq\ (1+\varepsilon)\sum_{i=1}^{k}\lambda_i(\mathbb{E}Y)\right\} \leq \binom{n}{k}^{\frac{1}{k}}\left(\frac{e^{\varepsilon}}{(1+\varepsilon)^{1+\varepsilon}}\right)^{\frac{1}{ck}\sum_{i=1}^{k}\lambda_i(\mathbb{E}Y)}, \quad \varepsilon \geq 0,$$

$$\mathbb{P}\left\{\sum_{i=1}^{k}\lambda_{n-i+1}(Y)\ \leq\ (1-\varepsilon)\sum_{i=1}^{k}\lambda_{n-i+1}(\mathbb{E}Y)\right\} \leq \binom{n}{k}^{\frac{1}{k}}\left(\frac{e^{-\varepsilon}}{(1-\varepsilon)^{1-\varepsilon}}\right)^{\frac{1}{ck}\sum_{i=1}^{k}\lambda_{n-i+1}(\mathbb{E}Y)}, \quad \varepsilon \in [0,1),$$

where $\lambda_i(Y)$ denotes the $i_{\text{th}}$ largest eigenvalue of $Y$ so $\sum_{i=1}^{k}\lambda_{n-i+1}(Y)$ is the sum of the smallest $k$ eigenvalues of $Y$. Our result shows that, the sum of the $k$ largest (or smallest) eigenvalues obeys a similar concentration law as the largest (or smallest) eigenvalue.

**Lieb's Concavity Theorem**

Tropp's proof of his concentration inequalities on extreme eigenvalues is based on the matrix Laplace transform method (see, for example, [74]) and also relies critically on a concavity theorem by Lieb: the function

$$A \ \longmapsto\ \mathrm{Tr}[\exp(H+\log A)], \tag{1.2}$$

is concave on $\mathbf{H}_n^{++}$, for any $n \times n$ Hermitian matrix $H$. Here $\mathbf{H}_n^{++}$ denotes the convex cone of all Hermitian, PD matrices. Note that this part of theories does not restrict to real symmetric matrices, hence we will be dealing with Hermitian matrices. The concavity of (1.2) is equivalent to a celebrated result in the study of trace inequalities, the joint concavity of the function

$$(A,B)\ \longmapsto\ \mathrm{Tr}[K^*A^pKB^q] \tag{1.3}$$

on $\mathbf{H}_n^+ \times \mathbf{H}_m^+$, for any $K \in \mathbb{C}^{n\times m}$, $p,q \in [0,1], p+q \leq 1$, known as Lieb's Concavity Theorem [67]. Here $\mathbf{H}_n^+$ is the convex cone of all $n \times n$ Hermitian, positive semidefinite matrices. This theorem answered affirmatively an important conjecture by Wigner, Yanase and Dyson [131] in information theory. It also led to Lieb's three-matrix extension of the Golden–Thompson inequality

$$\mathrm{Tr}\left[e^{A+B+C}\right] \leq \mathrm{Tr}\left[e^{A}\int_0^{\infty}(e^{-B}+tI)^{-1}e^{C}(e^{-B}+tI)^{-1}dt\right] \tag{1.4}$$

for arbitrary Hermitian matrices $A, B, C$ of the same size, which was then used by Lieb and Ruskai [68] to prove the strong subadditivity of quantum entropy.

In this part of work, we will generalize Lieb's concavity theorem from trace to a class of homogeneous matrix functions. In particular, we will prove that the function

$$(A, B) \longmapsto \mathrm{Tr}_k\left[(B^{\frac{q}{2}}K^*A^pKB^{\frac{q}{2}})^s\right]^{\frac{1}{k}} \tag{1.5}$$

is jointly concave on $\mathbf{H}_n^+ \times \mathbf{H}_m^+$, for any $K \in \mathbb{C}^{n\times m}$ and any $p, q \in [0, 1], s \in [0, \frac{1}{p+q}]$. The $k$**-trace** function $\mathrm{Tr}_k[A]$ of a square matrix $A$ denotes the $k_{\text{th}}$ elementary symmetric polynomial of the eigenvalues of $A$:

$$\mathrm{Tr}_k[A] = \sum_{1\le i_1<i_2<\cdots<i_k\le n} \lambda_{i_1}\lambda_{i_2}\cdots\lambda_{i_k}, \qquad 1 \le k \le n, \tag{1.6}$$

where $\lambda_1, \lambda_2, \cdots, \lambda_n$ are all eigenvalues of $A$. In particular, we have $\mathrm{Tr}_1[A] = \mathrm{Tr}[A]$ and $\mathrm{Tr}_n[A] = \det[A]$.

In the case $k = 1$, the concavity of function (1.5) has been studied by many and results with an increasing range of $s$ have been obtained over time: $1 \le s \le \frac{1}{p+q}$ [49] and $\frac{1}{2} \le s \le \frac{1}{p+q}$ [47] by Hiai, and $0 \le s \le \frac{1}{1+q}$ by Carlen, Frank and Lieb [25]. These partial results together already suffice to conclude the concavity for the full range $0 \le p, q \le 1, 0 \le s \le \frac{1}{p+q}$. The first complete proof of concavity covering the full range is due to Hiai [48]. Meanwhile, the convexity of function (1.5) with $k = 1$ for different ranges of $p, q, s$ has also been established. A complete convexity/concavity result for the full range of $p, q, s$ was recently accomplished by Zhang [136] using an elegant variational approach that is modified from a variational method developed by Carlen and Lieb [27]. Here "full" means the conditions are also necessary for the corresponding convexity/concavity to hold for all dimensions $n, m$. We refer to the papers [26] by Carlen, Frank and Lieb, and [136] by Zhang for historical overviews on this topic in more detail. In this thesis, we will only work on the concavity of function (1.5), since the map $A \mapsto \mathrm{Tr}_k[A]^{\frac{1}{k}}$ is homogeneous of order 1 and concave on $\mathbf{H}_n^+$ for $k \ge 2$. Our work therefore extends the established concavity results from the normal trace to a class of $k$-trace functions.

After establishing our generalized Lieb's concavity theorem, it is easy to further derive the concavity of the map

$$A \longmapsto \mathrm{Tr}_k\left[\exp(H + \log A)\right]^{\frac{1}{k}} \tag{1.7}$$

on $\mathbf{H}_n^{++}$, for any Hermitian matrix $H$ of the same size. This will then yield the desired concentration inequalities on sums of eigenvalues. Our proof of this part

is similar in spirit to Tropp's argument based on the original Lieb's theorem (1.2). However, the extension from trace to $k$-trace will endow this argument the power to control the partial sums of eigenvalues rather than just the extreme eigenvalues.

Since Lieb's original establishment of his concavity theorem, alternative proofs have been developed from different aspects of matrix theories, including matrix tensors (Ando [3], Carlen [24], Nikoufar et al. [85]), the theory of Herglotz functions (Epstein [37]), and interpolation theories (Uhlmann [124], Kosaki [58]). The tensor approaches prove the theorem elegantly by translating the concavity of (1.3) to the operator concavity of the map $(A, B) \mapsto A^p \otimes B^q$, but have difficulties in generalizing to our $k$-trace case due to the nonlinearity of $\mathrm{Tr}_k$. However, the $k$-trace has two good properties that are most essential to the desired results: (i) the map $A \mapsto \mathrm{Tr}_k[A]^{\frac{1}{k}}$ is concave on $\mathbf{H}_n^+$, and (ii) the $k$-trace satisfies Hölder's inequality as the normal trace does. We, therefore, turned to the more generalizable methods of operator interpolation based essentially on Hölder's inequality. Originating from the Hadamard three-lines theorem [45], the interpolation of operators has been a powerful tool in operator and functional analysis, with variants including the Riesz-Thorin interpolation theorem [97], Stein's interpolation of holomorphic operators [116], Peetre's K-method [92] and many others. In particular, we found Stein's complex interpolation technique most compatible and easiest to use in the $k$-trace setting. Our use of interpolation technique was inspired by a recent work of Sutter et al. [119], in which they applied Stein's interpolation to derive a multivariate extension of the Golden–Thompson inequality. This interpolation technique will help us first prove a key lemma that the function

$$A \longmapsto \mathrm{Tr}_k\left[(K^*A^pK)^s\right]^{\frac{1}{k}} \tag{1.8}$$

is concave on $\mathbf{H}_n^+$, for any $K \in \mathbb{C}^{n\times n}$ and any $p \in [0, 1]$, $s \in [0, \frac{1}{p}]$. Note that function (1.8) is a special case of function (1.5) with $q = 0$. Given this lemma, the concavity in the more general case can be obtained via a powerful variational argument that originates in [27] by Carlen and Lieb. This kind of variational methods, introducing supremum/infimum characterizations of trace functions, has been widely used in the study of the convexity/concavity of function (1.5) and its variants in the trace case (see, e.g., [24–27]). Our proof for the $k$-trace case is similar in inspirit to a refined variational approach by Zhang [136], which is again based on Hölder's inequalities.

## 1.4 Summary of the Thesis

The remaining thesis is organized as follows.

- In Chapter 2, we present the general framework of operator compression via energy decomposition. The overall approach consists of three major parts: (i) the construction of an adaptive partition $\mathcal{P}$, (ii) the choice of the coarse space $\Phi$ and its basis, and (iii) the construction of the modified coarse space $\Psi$ and its (localized) basis. We prove the spatially exponential decay property of the basis functions of $\Psi$, which ensures the nearly optimal complexity of our method. We introduce two local measurement factors, the error factor and the condition factor, to provide a theoretical guarantee (error estimate and complexity estimate) for our method and practical guidance for the design of our algorithms. The performance of our operator compression framework is then tested on a PDE example.

- In Chapter 3, we extend our operator compression framework to a MMD scheme. The development of our MMD relies on the inheritance of the energy decomposition through a pyramid-like hierarchical structure, from the finest level to the coarsest level. Our MMD scheme naturally leads to a parallelizable fast linear solver for large sparse PD linear systems. We illustrate by several graph Laplacian examples the effectiveness of our MMD approach in resolving the large condition number of a PD matrix by decomposing it into many well-conditioned pieces.

- In Chapter 4, we develop our MMD framework into a hierarchically preconditioned eigensolver for sparse positive semidefinite matrices, based on the celebrated scheme of the Implicit Restarted Lanczos Method. Our hierarchical eigensolver consists of two alternating procedures: the level-wise spectrum extension process that finds new eigenpair candidates, and the cross-level spectrum refinement process that refines the computed eigenspace. The extension process relies critically on the careful design of a spectral preserving preconditioner. The refinement process follows the Orthogonal Iteration with Ritz Acceleration which converges exponentially fast with properly chosen parameters. Our hierarchical eigensolver is tested on multiple numerical examples, which provide evidence that our method could outperform other existing methods.

- In Chapter 5, we prove some concentration inequalities on eigenvalue sums of random positive semidefinite matrices, extending existing concentration results on extreme eigenvalues. We introduce the notion of $k$-traces to provide bounds on sums of the largest (or smallest) eigenvalues. These new

concentration inequalities are consequences of a generalized Lieb's concavity theorem, which extends the famous Lieb's concavity theorem from normal trace to $k$-traces. The proof of our generalized Lieb's concavity theorem incorporates a variety of mathematical concepts and techniques, including the mixed discriminant, the exterior algebra, derivatives of matrix functions and operator interpolation.

*Chapter 2*

# OPERATOR COMPRESSION VIA ENERGY DECOMPOSITION

In this chapter, we introduce a novel framework of operator compression via energy decomposition. It provides a robust methodology for extracting the leading components of the inverse of a general class of positive definite (PD) matrix. Our approach consists of three major parts: the construction of an adaptive partition $\mathcal{P}$, the choice of the coarse space $\Phi$ and its basis, and the construction of the modified coarse space $\Psi$ and its (localized) basis. We provide rigorous compression error estimate and complexity estimate by introducing two local measurements which can be calculated efficiently by solving a local and partial eigen problem. These concepts and methods will be discussed under the following outline.

In Section 2.1, we will introduce the foundation of our work, which is the notion of **energy decomposition** of general PD matrices. Section 2.2 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 2.3. Discussion of the computational complexity is also included. To demonstrate the efficacy of our partitioning algorithm, two numerical results are reported in Section 2.4.

## 2.1 Energy Decomposition

We start by considering the linear system $Lx = b$, where $L$ is the Laplacian of an undirected, positive-weighted graph $\boldsymbol{G} = \{\boldsymbol{V}; \boldsymbol{E}, \boldsymbol{W}\}$, i.e.

$$L_{ij} = \begin{cases} \sum_{(i,j')\in\boldsymbol{E}} w_{ij'} & \text{if } i = j; \\ -w_{ij} & \text{if } i \neq j \text{ and } (i,j) \in \boldsymbol{E}; \\ 0 & \text{otherwise.} \end{cases} \tag{2.1}$$

We allow for the existence of self-loops $(i,i) \in \boldsymbol{E}$. When $L$ is singular we mean to solve $x = L^{\dagger}b$, where $L^{\dagger}$ is the pseudo-inverse of $L$. Our algorithm will base 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 path-wise information of the graph $\boldsymbol{G}$ by operating on each single edge in $\boldsymbol{E}$, which can be easily adapted to a larger class of linear systems with symmetric, positive semidefinite matrix. Notice that the contribution of an edge $(i, j) \in \boldsymbol{E}$ with weight $w_{ij}$ to the Laplacian matrix $L$ is simply

$$E_{ii} \triangleq \begin{array}{c} \begin{matrix} \quad & i & \quad \end{matrix} \\ \begin{pmatrix} 0 & & \\ & w_{ii} & \\ & & 0 \end{pmatrix} \end{array} \begin{matrix} \\ i \\ \\ \end{matrix}, \; i = j; \quad E_{ij} \triangleq \begin{array}{c} \begin{matrix} \quad & i & \quad & j & \quad \end{matrix} \\ \begin{pmatrix} 0 & & & & \\ & w_{ij} & & -w_{ij} & \\ & & \ddots & & \\ & -w_{ij} & & w_{ij} & \\ & & & & 0 \end{pmatrix} \end{array} \begin{matrix} \\ i \\ \\ j \\ \\ \end{matrix}, \; i \neq j. \tag{2.2}$$

And we have $L = \sum_{(i,j)\in\boldsymbol{E}} E_{ij}$. 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 $E_k \succeq 0$. Here $E_k \succeq 0$ means $E_k$ is positive semidefinite. Therefore, we will theoretically develop our method for general decomposable PD matrices. Also we assume that $A$ is invertible, as we can easily generalize our method to the case when $A^{\dagger}b$ is pursued.

We therefore 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 $n\times n$ symmetric, positive definite matrix. We define the **energy decomposition** as follows:

**Definition 2.1.1** (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*

$$A = \sum_{k=1}^{m} E_k, \quad E_k \succeq 0 \; \forall k = 1, \ldots, m. \tag{2.3}$$

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 $\mathcal{E}_{fine} = \{E_k\}_{k=2}^m \cup \{E_{k,1}, E_{k,2}\}$ 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 $Ev \neq \mathbf{0}$, since $E \succeq \mathbf{0}$ );
- For any $u, v \in \mathcal{V}$, we denote $u \sim v$ if $\exists E \in \mathcal{E}$ such that $u^T E v \neq 0$.

As an immediate example, if we take $\mathcal{V}$ to be the set of all distinct eigen vectors of $A$, then $v \nsim u$ 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 eigen vectors. 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 $\mathcal{V}, \mathcal{E}$ from graph perspective. Indeed, one can keep in mind that $G = \{\mathcal{V}, \mathcal{E}\}$ is the generalized concept of undirected graphs, where $\mathcal{V}$ stands for the set of vertices, and $\mathcal{E}$ stands for the set of edges. For any vertices (basis) $v, u \in \mathcal{V}$, and any edge (energy) $E \in \mathcal{E}$, $v \sim E$ 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 "multi-vertex edge" $E$ can be viewed as a counterpart of edge weight in graph setting. Conversely, if the problem comes directly from a weighted graph, then one can naturally construct the sets $\mathcal{V}$ and $\mathcal{E}$ from the vertices and edges of the graph as we will see in Example 2.1.10.

**Definition 2.1.2** (Neighboring)**.** *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 $\mathbb{R}^n$. For any $E \in \mathcal{E}$, We define $\mathcal{N}(E;\mathcal{V}) := \{v \in \mathcal{V} : E \sim v\}$ to be the set of $v \in \mathcal{V}$ **neighboring** $E$. Similarly, for any $v \in \mathcal{V}$, we define $\mathcal{N}(v;\mathcal{E}) := \{E \in \mathcal{E} : E \sim v\}$ and $\mathcal{N}(v) := \{u \in \mathcal{V} : u \sim v\}$ to be the set of $E \in \mathcal{E}$ and the set of $u \in \mathcal{V}$ **neighboring** $v \in \mathcal{V}$ respectively. Furthermore, for any $\mathcal{S} \subset \mathcal{V}$ and any $E \in \mathcal{E}$, we denote $E \sim \mathcal{S}$ if $\mathcal{N}(E;\mathcal{V}) \cap \mathcal{S} \neq \emptyset$ and $E \in \mathcal{S}$ if $\mathcal{N}(E;\mathcal{V}) \subset \mathcal{S}$.*

In what follows, we will see that if two energy elements $E_k, E_{k'}$ have the same neighbor basis, namely $\mathcal{N}(E_k;\mathcal{V}) = \mathcal{N}(E_{k'};\mathcal{V})$, 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 $\mathcal{N}(E_{k,1};\mathcal{V}) \subsetneq \mathcal{N}(E_k;\mathcal{V})$ or $\mathcal{N}(E_{k,2};\mathcal{V}) \subsetneq \mathcal{N}(E_k;\mathcal{V})$. 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 $\mathcal{S} \subset \mathcal{V}$ as follows:

**Definition 2.1.3** (Restricted, Interior and Closed energy)**.** *Let $\mathcal{E} = \{E_k\}_{k=1}^m$ be a energy decomposition of $A$. Let $\mathcal{S}$ be a subset of $\mathcal{V}$, and $P_{\mathcal{S}}$ be the orthogonal projection onto $\mathcal{S}$. The **restricted energy** of $\mathcal{S}$ with respect to $A$ is defined as*

$$A_{\mathcal{S}} := P_{\mathcal{S}} A P_{\mathcal{S}}; \tag{2.4}$$

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

$$\underline{A}_{\mathcal{S}}^{\mathcal{E}} = \sum_{E \in \mathcal{S}} E; \tag{2.5}$$

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

$$\overline{A}_{\mathcal{S}}^{\mathcal{E}} = \sum_{E \in \mathcal{S}} E + \sum_{E \notin \mathcal{S}, E \sim S} P_{\mathcal{S}} E^d P_{\mathcal{S}}, \tag{2.6}$$

*where*

$$E^d = \sum_{v \in \mathcal{V}} \Big( \sum_{u \in \mathcal{V}} |v^T E u| \Big) v v^T = \sum_{v \sim E} \Big( \sum_{u \sim v} |v^T E u| \Big) v v^T \tag{2.7}$$

*is called the* ***diagonal concentration*** *of $E$, and we have*

$$P_{\mathcal{S}} E^d P_{\mathcal{S}} = \sum_{v\in\mathcal{S}, v\sim E} \Big( \sum_{u\sim v} |v^T E u| \Big) v v^T \tag{2.8}$$

**Remark 2.1.4.** *The restricted energy of $\mathcal{S}$ can be simply viewed as the restriction of $A$ on the subset $\mathcal{S}$. The interior energy (closed energy) of $\mathcal{S}$ is $A_{\mathcal{S}}$ excluding (including) contributions from other energy elements $E \notin \mathcal{S}$ neighboring $\mathcal{S}$. The following example illustrates the idea of various energies introduced in Definition 2.1.3 by considering the 1-dimensional discrete Laplace operator with Dirichlet boundary conditions.*

**Example 2.1.5.** *Consider $A$ to be the $(n+1)\times(n+1)$ tridiagonal matrix with entries -1 and 2 on off-diagonals and diagonal respectively. Let*

$$E_1 = \begin{pmatrix} 2 & -1 & \\ -1 & 1 & \\ & & 0 \end{pmatrix},\ E_n = \begin{pmatrix} 0 & & \\ & 1 & -1 \\ & -1 & 2 \end{pmatrix},\ \text{and } E_k = \begin{pmatrix} 0 & & \\ & 1 & -1 \\ & -1 & 1 \\ & & & 0 \end{pmatrix} \tag{2.9}$$

*for $k = 2, \dots n-1$. Let $\mathcal{V} = \{\boldsymbol{e}_i\}_{i=0}^n$ to 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 $\mathcal{S} = \{\boldsymbol{e}_3, \boldsymbol{e}_4, \boldsymbol{e}_5, \boldsymbol{e}_6\}$, then we have*

$$A_{\mathcal{S}} = \begin{pmatrix} 0 & & & & \\ & 2 & -1 & & \\ & -1 & 2 & -1 & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \\ & & & & & 0 \end{pmatrix}, \quad \underline{A}_{\mathcal{S}}^{\mathcal{E}} = \begin{pmatrix} 0 & & & & \\ & 1 & -1 & & \\ & -1 & 2 & -1 & \\ & & -1 & 2 & -1 \\ & & & -1 & 1 \\ & & & & & 0 \end{pmatrix},$$

$$\text{and} \quad \overline{A}_{\mathcal{S}}^{\mathcal{E}} = \begin{pmatrix} 0 & & & & \\ & 3 & -1 & & \\ & -1 & 2 & -1 & \\ & & -1 & 2 & -1 \\ & & & -1 & 3 \\ & & & & & 0 \end{pmatrix}.$$

*Recall that the interior energy $\underline{A}_{\mathcal{S}}^{\mathcal{E}} = \sum_{E_k\in\mathcal{S}} E_k = \sum_{k=4}^{6} E_k$, while the closed energy*

$$\begin{aligned} \overline{A}_{\mathcal{S}}^{\mathcal{E}} &= \underline{A}_{\mathcal{S}}^{\mathcal{E}} + \sum_{E\notin\mathcal{S}, E\sim\mathcal{S}} P_{\mathcal{S}} E^d P_{\mathcal{S}} \\ &= \underline{A}_{\mathcal{S}}^{\mathcal{E}} + |\boldsymbol{e}_3^T E_3 \boldsymbol{e}_2| \boldsymbol{e}_3 \boldsymbol{e}_3^T + |\boldsymbol{e}_3^T E_3 \boldsymbol{e}_3| \boldsymbol{e}_3 \boldsymbol{e}_3^T + |\boldsymbol{e}_7^T E_7 \boldsymbol{e}_6| \boldsymbol{e}_6 \boldsymbol{e}_6^T + |\boldsymbol{e}_6^T E_7 \boldsymbol{e}_6| \boldsymbol{e}_6 \boldsymbol{e}_6^T \end{aligned}$$

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

**Remark 2.1.6.**

- *Notice that any eigenvector $x$ of $A_{\mathcal{S}}$ (or $\underline{A}_{\mathcal{S}}^{\mathcal{E}}$, $\overline{A}_{\mathcal{S}}^{\mathcal{E}}$) corresponding to nonzero eigenvalue must satisfy $x \in \operatorname{span}(\mathcal{S})$. In this sense, we also say $A_{\mathcal{S}}$ (or $\underline{A}_{\mathcal{S}}^{\mathcal{E}}$, $\overline{A}_{\mathcal{S}}^{\mathcal{E}}$) is local to $\mathcal{S}$.*

- *For any energy $E$, we have $E \preceq E^d$ since for any $x = \sum_{i=1}^n c_i v_i$, we have*

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

**Proposition 2.1.7.** *For any $\mathcal{S} \subset \mathcal{V}$, we have that $\underline{A}_{\mathcal{S}}^{\mathcal{E}} \preceq A_{\mathcal{S}} \preceq \overline{A}_{\mathcal{S}}^{\mathcal{E}}$.*

*Proof.* We have

$$\underline{A}_{\mathcal{S}}^{\mathcal{E}} = \sum_{E \in \mathcal{S}} E \preceq \sum_{E \in \mathcal{S}} E + \sum_{\substack{E \notin \mathcal{S}, \\ E \sim S}} P_{\mathcal{S}} E P_{\mathcal{S}} \preceq \sum_{E \in \mathcal{S}} E + \sum_{\substack{E \notin \mathcal{S}, \\ E \sim S}} P_{\mathcal{S}} E^d P_{\mathcal{S}} = \overline{A}_{\mathcal{S}}^{\mathcal{E}}.$$

Notice that $P_{\mathcal{S}} E P_{\mathcal{S}} = E$ for $E \in \mathcal{S}$, and $P_{\mathcal{S}} E P_{\mathcal{S}} = 0$ for $E \nsim \mathcal{S}$, thus

$$A_{\mathcal{S}} = P_{\mathcal{S}} A P_{\mathcal{S}} = \sum_{E \in \mathcal{S}} E + \sum_{\substack{E \notin \mathcal{S}, \\ E \sim S}} P_{\mathcal{S}} E P_{\mathcal{S}},$$

and the desired result follows. □

**Definition 2.1.8** (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 2.1.9.** *Let $\mathcal{E} = \{E_k\}_{k=1}^m$ be an energy decomposition of $A$, and $\mathcal{P} = \{P_j\}_{j=1}^M$ be a partition of $\mathcal{V}$. Then*

$$\sum_{j=1}^M \underline{A}_{P_j}^{\mathcal{E}} \preceq A \preceq \sum_{j=1}^M \overline{A}_{P_j}^{\mathcal{E}}. \tag{2.10}$$

*Proof.* Let $\mathcal{E}_{\mathcal{P}} = \{E \in \mathcal{E} : \exists P_j \in \mathcal{P} \text{ such that } E \in P_j\}$, and $\mathcal{E}^c_{\mathcal{P}} = \mathcal{E}\backslash\mathcal{E}_{\mathcal{P}}$. Recall that $E \in P_j$ if $\mathcal{N}(E, \mathcal{V}) \subset P_j$ (See Definition 2.1.2). 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 $\sum_j P_j = \mathbf{Id}$. Then

$$\begin{aligned}
\sum_j \underline{A}^{\mathcal{E}}_{P_j} &= \sum_j \sum_{E\in P_j} E \\
&\preceq \sum_j \sum_{E\in P_j} E + \sum_{E\in\mathcal{E}^c_{\mathcal{P}}} E \\
&\preceq \sum_j \sum_{E\in P_j} E + \sum_{E\in\mathcal{E}^c_{\mathcal{P}}} \Big[(\sum_j P_j) E^d (\sum_{j'} P_{j'})\Big] \\
&= \sum_j \sum_{E\in P_j} E + \sum_{E\in\mathcal{E}^c_{\mathcal{P}}} (\sum_j P_j E^d P_j) \\
&= \sum_j \Big(\sum_{E\in P_j} E + \sum_{E\notin P_j, E\sim P_j} P_j E^d P_j\Big) \\
&= \sum_j \overline{A}^{\mathcal{E}}_{P_j}.
\end{aligned}$$

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

$$A = \sum_{E\in\mathcal{E}_{\mathcal{P}}} E + \sum_{E\in\mathcal{E}^c_{\mathcal{P}}} E = \sum_j \sum_{E\in P_j} E + \sum_{E\in\mathcal{E}^c_{\mathcal{P}}} E,$$

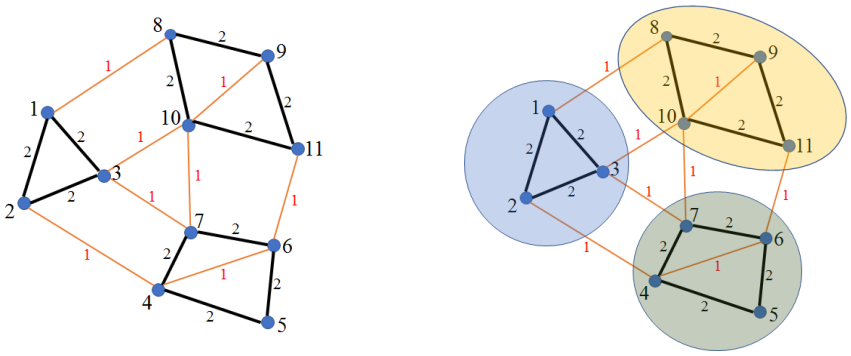
and the desired result follows. □

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

**Example 2.1.10.** *Consider L to be the graph Laplacian matrix of the graph given in Figure 2.1. 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 Equation* (2.2)*). Now suppose we have given the partition* $\mathcal{P} = \left\{P_j\right\}_{j=1}^3$ *with* $P_1 = [\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3]$, $P_2 = [\boldsymbol{e}_4, \boldsymbol{e}_5, \boldsymbol{e}_6, \boldsymbol{e}_7]$ *and* $P_3 = [\boldsymbol{e}_8, \boldsymbol{e}_9, \boldsymbol{e}_{10}, \boldsymbol{e}_{11}]$, *where* $\boldsymbol{e}_i$ *are the standard basis of* $\mathbb{R}^{11}$. *Then we can obtain* $\underline{L}_{P_j}$ *and* $\overline{L}_{P_j}$ *as follows:*

$$\underline{L}_{P_1} = \begin{pmatrix} 4 & -2 & -2 \\ -2 & 4 & -2 \\ -2 & -2 & 4 \end{pmatrix}_{P_1}, \ \underline{L}_{P_2} = \begin{pmatrix} 5 & -2 & -1 & -2 \\ -2 & 4 & -2 & 0 \\ -1 & -2 & 5 & 2 \\ -2 & 0 & -2 & 4 \end{pmatrix}_{P_2}, \ \underline{L}_{P_3} = \begin{pmatrix} 4 & -2 & -2 & 0 \\ -2 & 5 & -1 & -2 \\ -2 & -1 & 5 & -2 \\ 0 & -2 & -2 & 4 \end{pmatrix}_{P_3}$$

$$\overline{L}_{P_1} = \begin{pmatrix} 6 & -2 & -2 \\ -2 & 6 & -2 \\ -2 & -2 & 8 \end{pmatrix}_{P_1}, \ \overline{L}_{P_2} = \begin{pmatrix} 7 & -2 & -1 & -2 \\ -2 & 4 & -2 & 0 \\ -1 & -2 & 7 & 2 \\ -2 & 0 & -2 & 8 \end{pmatrix}_{P_2}, \ \overline{L}_{P_3} = \begin{pmatrix} 6 & -2 & -2 & 0 \\ -2 & 5 & -1 & -2 \\ -2 & -1 & 9 & -2 \\ 0 & -2 & -2 & 6 \end{pmatrix}_{P_3}$$



(a) (b)

Figure 2.1: (a) An illustration of a graph example . (b) An illustration of a partition $\mathcal{P} = \{\{1, 2, 3\}, \{4, 5, 6, 7\}, \{8, 9, 10, 11\}\}$.

*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.*

## 2.2 Operator Compression

As mentioned in the Section 1.1, inspired by the Finite Element Method (FEM) approach for solving partial differential equation (PDE) 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 $\Psi = \Phi - P_U^A\Phi$ in [75], or $\Psi = A^{-1}(\Phi)$ as proposed in [53, 88]. Here $U = (\Phi)^\perp$ is the $l_2$-orthogonal complement space and $P_U^A$ 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 symmetric, PD matrix $A$.

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 (2.11). The following theorem also gives us a general idea on how we can control the errors introduced during the compression of the operator $A$.

**Theorem 2.2.1.** *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*

$$\|x - P_\Phi x\|_2 \leq \epsilon \|x\|_A, \quad \forall x \in \mathbb{R}^n, \tag{2.11}$$

*for some $\epsilon > 0$, then*

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

$$\|x - P_\Psi^A x\|_A \leq \epsilon \|b\|_2. \tag{2.12}$$

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

$$\|x - P_\Psi^A x\|_2 \leq \epsilon^2 \|b\|_2. \tag{2.13}$$

3. *We have*

$$\|A^{-1} - P_\Psi^A A^{-1}\|_2 \leq \epsilon^2. \tag{2.14}$$

*Proof.* 1. Let $y = A^{-1}(P_\Phi b) \in \Psi$, then

$$\begin{aligned}\|x - y\|_A^2 &= (x - y, A(x - y)) = (x - y - P_\Phi(x - y), b - P_\Phi b) \\ &\leq \|x - y - P_\Phi(x - y)\|_2 \|b - P_\Phi b\|_2 \leq \epsilon \|x - y\|_A \|b\|_2,\end{aligned}$$

and thus we have

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

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

$$\begin{aligned}\|x - P_\Psi^A x\|_2^2 &= (x - P_\Psi^A x, x - P_\Psi^A x) = (x - P_\Psi^A x, Az) \\ &= (x - P_\Psi^A x, z - P_\Psi^A z)_A \leq \|x - P_\Psi^A x\|_A \|z - P_\Psi^A z\|_A \\ &\leq \epsilon \|x - P_\Psi^A x\|_A \|Az\|_2,\end{aligned}$$

and thus

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

3. Immediate result of 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

$$P_\Psi^A = \Psi(\Psi^T A\Psi)^{-1}\Psi^T A = A^{-1}\Phi(\Phi^T A^{-1}\Phi)^{-1}\Phi^T, \tag{2.15}$$

which is the $A$-orthogonal projection matrix into the subspace $\Psi$. The corresponding compressed approximation of $A^{-1}$ is given by

$$P_\Psi^A A^{-1} = \Psi(\Psi^T A\Psi)^{-1}\Psi^T. \tag{2.16}$$

In [75], Målqvist and Petersein 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 [75], then $\Psi$ is exactly the modified coarse space $V_H^{ms}$ as they proposed, and the first error estimate in Theorem 2.2.1 is consistent to their error analysis. More generally, Owhadi in [89] makes use of the Gamblet framework to construct the basis of modified coarse space such that the conforming properties of that basis is no longer required. In particular, (2.15) is an analogy of $\Psi\Phi = K\Psi^T(\Phi K\Phi^T)^{-1}\Phi$ in page 9 of [88] and the error estimate in (2.12) is corresponding to Proposition 3.6 in that paper.

As a 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) [55]. And the optimal compression error is given by $\epsilon^2 = (\lambda_{N+1}(A))^{-1}$. Though with 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 [34, 80, 137] provides a strategy to obtain a sparse approximation of $A^{-1}$, it implicitly assumes that the operators inherit a low rank characteristics such that $l_1$ minimization approach is effective. To fully use the local properties of $A$, we would prefer to choose $\Phi$ that can be locally computed but still has good approximation property, namely satisfying condition (2.11) 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) piece-wise polynomials with compact local supports, and the error

is given by the resolution of the partition of the computational domain [8, 19]. 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 have a priori the computational domain? Such 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 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 Section 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 $A_{\text{st}} = \Psi^T A \Psi$ inherited from $A$, and to give a reasonable bound on the condition number of $A_{\text{st}}$.

**Algorithm 1** *Operator Compression*

**Input:** Energy decomposition $\mathcal{E}$, underlying basis $\mathcal{V}$, desire accuracy $\epsilon$

1: Construct partition $\mathcal{P}$ subject to $\epsilon$ using Algorithm 4;
2: Construct $\Phi$ using Algorithm 2;
3: Construct $\widetilde{\Psi}$ using Algorithm 3 subject to $\epsilon$;
4: Compute $P^A_{\widetilde{\Psi}} A^{-1} = \widetilde{\Psi}(\widetilde{\Psi}^T A \widetilde{\Psi})^{-1} \widetilde{\Psi}^T$ as the compressed operator;

In summary, as mentioned in Section 1.1, 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 (2.11); (iii) construct $\Psi$ that provides stiffness matrix $A_{\text{st}}$ with locality and reasonable condition number. The whole process can be summarized as the Algorithm 1, and each step will be discussed in following sections.

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.

### 2.2.1 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 (2.11) 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 2.2.2.** *Let $\mathcal{P} = \{P_j\}_{j=1}^M$ be a partition of $\mathcal{V}$, and $\{\underline{A}_{P_j}\}_{j=1}^M$ be the corresponding interior energies as defined in Definition 2.1.3. For each $1 \le j \le M$, let $\Phi_j$ be some subspace of* $\operatorname{span}\{P_j\}$ *such that*

$$\|x - P_{\Phi_j} x\|_2 \le \epsilon \|x\|_{\underline{A}_{P_j}}, \quad \forall x \in \operatorname{span}\{P_j\}, \tag{2.17}$$

*for some constant $\epsilon$. Then we have*

$$\|x - P_\Phi x\|_2 \le \epsilon \|x\|_A, \quad \forall x \in \mathbb{R}^n, \tag{2.18}$$

*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

$$\|x - P_\Phi x\|_2^2 = \|\sum_j (P_j x - P_{\Phi_j} x)\|_2^2 = \sum_j \|P_j x - P_{\Phi_j} x\|_2^2 \le \epsilon^2 \sum_j \|P_j x\|_{\underline{A}_{P_j}}^2.$$

Notice that

$$\sum_j \|P_j x\|_{\underline{A}_{P_j}}^2 = \sum_j \|x\|_{\underline{A}_{P_j}}^2 = \|x\|_{\underline{A}}^2 \le \|x\|_A^2,$$

and the conclusion follows. □

Intuitively, given a partition $\mathcal{P}$ of $\mathcal{V}$, we can construct $\Phi$ locally by choosing $\Phi_j$ that satisfies (2.17) 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 (2.17). 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 $\mathcal{P} = \{P_j\}_{j=1}^M$. Then 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 the interior energy $A_S$ for a subset $\mathcal{S} \subset \mathcal{V}$. Lemma 2.2.4 will then show the relationship between the interior spectrum and the minimum dimension that can be achieved for each $\Phi_j$.

**Definition 2.2.3** (Interior Spectrum)**.** *Let $\mathcal{S}$ be a subset of $\mathcal{V}$. We define the interior spectrum $\{\lambda_j(\mathcal{S}, A)\}_{j=1}^{s}$ as the set of eigenvalues of $\underline{A}_{\mathcal{S}}$, where $s = \#\mathcal{S} = \dim(\mathrm{span}\{\mathcal{S}\})$ and $\underline{A}_{\mathcal{S}}$ is the interior energy of $\mathcal{S}$ with respect to $A$ (as an operator restricted to the space $\mathrm{span}\{\mathcal{S}\}$).*

In what follows, since $A$ is generally given and fixed, we will write $\lambda_j(\mathcal{S}; A)$ as $\lambda_j(\mathcal{S})$. Also we will always assume the ordering $\lambda_1(\mathcal{S}) \le \lambda_2(\mathcal{S}) \le \cdots \le \lambda_s(\mathcal{S})$.

**Lemma 2.2.4.** *Given a set $\mathcal{S} \subset \mathcal{V}$ and a constant $\epsilon$, let $q(\epsilon)$ be the smallest integer such that $\frac{1}{\epsilon^2} \le \lambda_{q(\epsilon)+1}(\mathcal{S})$. Also define $\mathcal{G}(\epsilon) = \{\Theta \subset \mathrm{span}\{\mathcal{S}\} : \|x - P_\Theta x\|_2 \le \epsilon \|x\|_{\underline{A}_{\mathcal{S}}},\ \forall x \in \mathrm{span}\{\mathcal{S}\}\}$, and let $p(\epsilon) = \min_{\Theta \in \mathcal{G}(\epsilon)} \dim \Theta$. Then we have $q(\epsilon) = p(\epsilon)$.*

*Proof.* Let $\Phi^k \subset \mathrm{span}\{\mathcal{S}\}$ denote the eigenspace of $\underline{A}_{\mathcal{S}}$ (as an operator restricted to $\mathrm{span}\{\mathcal{S}\}$) corresponding to interior eigenvalues $\lambda_1(\mathcal{S}) \le \lambda_2(\mathcal{S}) \le \cdots \le \lambda_k(\mathcal{S})$. On the one hand, for all $x \in \mathrm{span}\{\mathcal{S}\}$, we have

$$\begin{aligned}\|x\|^2_{\underline{A}_{\mathcal{S}}} &\ge \|x - P_{\Phi^{q(\epsilon)}} x\|^2_{\underline{A}_{\mathcal{S}}} \\ &= (x - P_{\Phi^{q(\epsilon)}} x)^T \underline{A}_{\mathcal{S}} (x - P_{\Phi^{q(\epsilon)}} x) \ge \lambda_{q(\epsilon)+1}(\mathcal{S}) \|x - P_{\Phi^{q(\epsilon)}} x\|_2^2,\end{aligned}$$

which is

$$\|x - P_{\Phi^{q(\epsilon)}} x\|_2 \le \frac{1}{\sqrt{\lambda_{q(\epsilon)+1}(\mathcal{S})}} \|x\|_{\underline{A}_{\mathcal{S}}} \le \epsilon \|x\|_{\underline{A}_{\mathcal{S}}}.$$

Thus $\Phi^{q(\epsilon)} \in \mathcal{G}(\epsilon)$, $q(\epsilon) \ge p(\epsilon)$. On the other hand, assume that the minimum $p(\epsilon)$ is achieved by some space $\widetilde{\Theta}$, then one can check that

$$\begin{aligned}\lambda_{p(\epsilon)+1} &= \max_{\substack{\Theta \subset \mathrm{span}\{\mathcal{S}\} \\ \dim \Theta = p(\epsilon)}} \min_{x \in \mathrm{span}\{\mathcal{S}\}} \frac{\|x - P_\Theta x\|^2_{\underline{A}_{\mathcal{S}}}}{\|x - P_\Theta x\|_2^2} \\ &\ge \min_{x \in \mathrm{span}\{\mathcal{S}\}} \frac{\|x - P_{\widetilde{\Theta}} x\|^2_{\underline{A}_{\mathcal{S}}}}{\|x - P_{\widetilde{\Theta}} x\|_2^2} \\ &= \min_{x \in \mathrm{span}\{\mathcal{S}\}} \frac{\|x - P_{\widetilde{\Theta}} x\|^2_{\underline{A}_{\mathcal{S}}}}{\|x - P_{\widetilde{\Theta}} x - P_{\widetilde{\Theta}}(x - P_{\widetilde{\Theta}} x)\|_2^2} \qquad \left(\text{since } P_{\widetilde{\Theta}}(x - P_{\widetilde{\Theta}} x) = 0\right) \\ &= \min_{\substack{y = x - P_{\widetilde{\Theta}} x \\ x \in \mathrm{span}\{\mathcal{S}\}}} \frac{\|y\|^2_{\underline{A}_{\mathcal{S}}}}{\|y - P_{\widetilde{\Theta}} y\|_2^2} \\ &\ge \min_{y \in \mathrm{span}\{\mathcal{S}\}} \frac{\|y\|^2_{\underline{A}_{\mathcal{S}}}}{\|y - P_{\widetilde{\Theta}} y\|_2^2} \ge \frac{1}{\epsilon^2},\end{aligned}$$

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

By Lemma 2.2.4, one optimal way to minimize $\dim \Phi_j$ for each $P_j$ subject to condition (2.17), is to take $\Phi_j = \Phi_j^{q_j(\epsilon)}$, the eigenspace corresponding to interior eigenvalues $\lambda_1(P_j) \leq \lambda_2(P_j) \leq \cdots \leq \lambda_{q_j(\epsilon)}(P_j)$, where $q_j(\epsilon)$ is the smallest integer such that $\frac{1}{\epsilon^2} \leq \lambda_{q_j(\epsilon)+1}(P_j)$. 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 $P_j$. Then the problem of minimizing $\dim \Phi$ subject to the condition (2.17) is reduced to finding a partition $\mathcal{P} = \{P_j\}_{j=1}^M$ with a minimal patch number. Still guided by Lemma 2.2.4, we know that we should choose $\Phi_j = \Phi_j^q$, and the condition (2.17) is satisfied if and only if $\frac{1}{\epsilon^2} \leq \lambda_{q+1}(P_j)$ for each $P_j$.

**Definition 2.2.5** (Error factor)**.** *Let $\mathcal{P} = \{P_j\}_{j=1}^M$ be a partition of $\mathcal{V}$. The **error factors** of $\mathcal{P}$ are defined as*

$$\varepsilon(P_j, q) = \frac{1}{\sqrt{\lambda_{q+1}(P_j)}}, \quad 1 \leq j \leq M, \quad \text{and} \quad \varepsilon(\mathcal{P}, q) = \max_j \frac{1}{\sqrt{\lambda_{q+1}(P_j)}}. \tag{2.19}$$

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

**Construction 2.2.6** (Construction of $\Phi$)**.** *We choose $\Phi = \bigoplus_{j=1}^M \Phi_j^q$, where $\Phi_j^q \subset \operatorname{span}\{P_j\}$ 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* (2.17) *is satisfied if $\varepsilon(\mathcal{P}, q) \leq \epsilon$.*

Guided by Construction 2.2.6, we propose Algorithm 2 to construct $\Phi$. Notice that it also computes the compliment space $U_j$ of $\Phi_j$ in each $\operatorname{span}(P_j)$, which will serve for the purpose of performing MMD in Section 3.1.

**Remark 2.2.7.**

- *The construction of $U_j$ can be implicitly done, for example, by extending $\Phi_j$ to an orthonormal basis of* $\mathrm{span}(P_j)$ *with local QR factorization, where only $q$ Householder vectors $[h_1, h_2, \cdots, h_q]$ need to be stored. In fact, we can apply economic QR factorization to $\Phi_j$ to obtain $(I - h_1 h_1^T)(I - h_2 h_2^T) \cdots (I - h_q h_q^T) = [Q_j, U_j]$ where $[Q_j, U_j]$ is orthogonal and* $\mathrm{Span}(\Phi_j) = \mathrm{Span}(Q_j)$. *In 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 $y = (I - h_q h_q^T) \cdots (I - h_2 h_2^T)(I - h_1 h_1^T)x$ and then taking the last $s - q$ entries of $y$; the latter one can be done by extending $x$ to $\tilde{x} = [\mathbf{0}, x]$ with additional $q$ 0s in front and then computing $(I - h_1 h_1^T)(I - h_2 h_2^T) \cdots (I - h_q h_q^T)\tilde{x}$.*

- *The integer $q$ is given before the partition is constructed, and the choice of $q$ will be discussed in Section 2.3.*

**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 $\underline{A}_{P_j}$ from $\mathcal{E}$;
3: Find the first $q$ normalized eigenvectors of $\underline{A}_{P_j}$ as $\Phi_j$;
4: Find $U_j$ such that $[\Phi_j, U_j]$ is an orthonormal basis of $\mathrm{span}(P_j)$;
5: **end for**
6: Collect all $\Phi_j$ as $\Phi$, and all $U_j$ as $U$.

**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 patch-wise complexity of solving partial eigen problem and extending $\Phi_j$ to $[\Phi_j, U_j]$. Then the complexity of Algorithm 2 is

$$O(\frac{F(s)}{s} \cdot n). \tag{2.20}$$

### 2.2.2 Choice of $\Psi$

Suppose that we have determined the space $\Phi = [\varphi_1, \varphi_2, \cdots, \varphi_N]$, the next step is to find $\Psi = [\psi_1, \psi_2, \cdots, \psi_N] = A^{-1}\Phi T$, namely to determine $T$, so that

1. each $\psi_i$ is locally computable, or can be approximated by some $\tilde{\psi}_i$ that is locally computable;

2. the stiffness matrix $A_{\text{st}} = \Psi^T A \Psi$ has relatively small condition number, or the condition number can be bounded by some local information.

Generally each $A^{-1}\phi_i$ 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 $\psi$ that can be well approximated by some $\tilde{\psi}_i$ which is locally computable.

**Lemma 2.2.8.** *Assume that* $\Psi = [\psi_1, \psi_2, \cdots, \psi_N]$ *satisfies* $\|x - P^A_\Psi x\|_A \leq \epsilon \|Ax\|_2$ *and* $\|A^{-1}_{st}\|_2 \leq \|A^{-1}\|_2$, *and that* $\widetilde{\Psi} = [\tilde{\psi}_1, \tilde{\psi}_2, \cdots, \tilde{\psi}_N]$ *satisfies* $\|\psi_i - \tilde{\psi}_i\|_A \leq \frac{C\epsilon}{\sqrt{N}}$, $1 \leq i \leq N$ *for some constant* $C$. *Then we have*

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

$$\|x - P^A_{\widetilde{\Psi}} x\|_A \leq (1 + C\|A^{-1}\|_2)\epsilon \|b\|_2.$$

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

$$\|x - P^A_{\widetilde{\Psi}} x\|_2 \leq (1 + C\|A^{-1}\|_2)^2 \epsilon^2 \|b\|_2.$$

3. *We have*

$$\|A^{-1} - P^A_{\widetilde{\Psi}} A^{-1}\|_2 \leq (1 + C\|A^{-1}\|_2)^2 \epsilon^2.$$

*Proof.* We only need to prove property 1, properties 2 and 3 follow by using the same argument as in Theorem 2.2.1. Recall that we have

$$\|x - P^A_\Psi x\|_A = \|x - \Psi c\|_A \leq \epsilon \|b\|_2,$$

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

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

Notice that

$$c^T c = x^T A \Psi A^{-2}_{\text{st}} \Psi^T A x \leq \|A^{\frac{1}{2}} \Psi A^{-2}_{\text{st}} \Psi^T A^{\frac{1}{2}}\|_2 \|x\|^2_A,$$
$$\|A^{\frac{1}{2}} \Psi A^{-2}_{\text{st}} \Psi^T A^{\frac{1}{2}}\|_2 = \|A^{-1}_{\text{st}} \Psi^T A \Psi A^{-1}_{\text{st}}\|_2 = \|A^{-1}_{\text{st}}\|_2,$$
$$\|x\|^2_A = b^T A^{-1} b \leq \|A^{-1}\|_2 \|b\|^2_2,$$

therefore we get

$$\|y_1 - y_2\|_A \leq C\epsilon \sqrt{\|A^{-1}_{\text{st}}\|_2 \|A^{-1}\|_2} \|b\|_2 \leq C\epsilon \|A^{-1}\|_2 \|b\|_2.$$

Then we have

$$\|x-y_2\|_A \le \|x-y_1\|_A+\|y_1-y_2\|_A \le \epsilon\|b\|_2+C\epsilon\|A^{-1}\|_2\|b\|_2 = (1+C\|A^{-1}\|_2)\epsilon\|b\|_2.$$

Since $y_2 \in \mathrm{span}\{\widetilde{\Psi}\}$, we obtain

$$\|x - P^A_{\widetilde{\Psi}}x\|_A \le \|x - y_2\|_A \le (1 + C\|A^{-1}\|_2)\epsilon\|b\|_2.$$

□

Guided by Lemma 2.2.8, 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 $\psi_i \in A^{-1}(\Phi)$ so that $\|\psi_i - \tilde{\psi}_i\|_A$ is small for each $i$. Since $\tilde{\psi}_i$ is given, minimizing $\|\psi_i - \tilde{\psi}_i\|_A$ can be simply solved by taking $\psi_i = P^A_\Psi\tilde{\psi}_i$. Thanks to the expression $P^A_\Psi = A^{-1}\Phi(\Phi^TA^{-1}\Phi)^{-1}\Phi^T$, we can perform the energy projection $P^A_\Psi$ as long as we know $\Phi$. Therefore we have

$$\Psi = P^A_\Psi\widetilde{\Psi} = A^{-1}\Phi(\Phi^TA^{-1}\Phi)^{-1}\Phi^T\widetilde{\Psi}, \tag{2.21}$$

$$\Longrightarrow \Phi^T\Psi = \Phi^TA^{-1}\Phi(\Phi^TA^{-1}\Phi)^{-1}\Phi^T\widetilde{\Psi} = \Phi^T\widetilde{\Psi}. \tag{2.22}$$

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

$$\varphi^T_{i'}\tilde{\psi}_i = \delta_{i',i}, \quad 1 \le i' \le N, \quad i.e. \quad \Phi^T\Psi = \Phi^T\widetilde{\Psi} = I_N. \tag{2.23}$$

Adding this extra localization constraint to the form of $\Psi = A^{-1}\Phi T$, we can choose $\Psi$ as follows:

**Construction 2.2.9** (Construction of $\Psi$)**.** *We choose $\Psi = A^{-1}\Phi T$ so that $\Phi^T\Psi = I_N$, that is*

$$\Psi = A^{-1}\Phi(\Phi^TA^{-1}\Phi)^{-1}, \quad T = (\Phi^TA^{-1}\Phi)^{-1}, \tag{2.24}$$

*and we have*

$$A_{st} = \Psi^T A \Psi = (\Phi^T A^{-1} \Phi)^{-1}. \tag{2.25}$$

**Remark 2.2.10.** *Our choice of $\Psi$ is inspired by the result proposed by Owhadi in [88], where the author obtained the same format of $\Psi$ 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 $\Psi$. Our derivation of the choice of $\Psi$ can be seen as a algebraic interpretation of Owhadi's probabilistic construction.*

Though we construct each $\psi_i$ from some local vector $\tilde{\psi}_i$, the error $\|\psi_i - \tilde{\psi}_i\|_A$ 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 [88]) shows that the construction of $\Psi$ in (2.24) is equivalent to the optimizer of a minimization problem.

**Lemma 2.2.11.** *Let $\Psi$ be constructed as in* (2.24)*. Then for each $i$, $\psi_i$ satisfies*

$$\begin{aligned} \psi_i &= \arg\min_{x \in \mathbb{R}^n} \|x\|_A, \\ \textit{subject to} \quad \varphi_{i'}^T x &= \delta_{i',i}, \quad \forall\, i' = 1, \ldots, N. \end{aligned}$$

*Proof.* Notice that $A\psi_i \in \Phi$, thus for any $x$ that satisfies $\varphi_{i'}^T x = \delta_{i',i}$, we have $\Phi^T(x - \psi_i) = 0$, and $\psi_i^T A(x - \psi_i) = 0$. Then we have

$$\|x\|_A^2 = \|x - \psi_i + \psi_i\|_A^2 = \|\psi_i\|_A^2 + \|x - \psi_i\|_A^2 + 2\psi_i^T A(x - \psi_i) \geq \|\psi_i\|_A^2. \tag{2.26}$$

□

By the construction given in (2.24) and guided by Lemma 2.2.11, we can obtain every $\psi_i$ by solving the optimization problem. Our next step is to make use of this minimal property to construct local $\tilde{\psi}_i$ that will be proved exponentially convergent to $\psi_i$.

**Definition 2.2.12** (Layers of neighbors)**.** *Let $\mathcal{P} = \{P_j\}_{j=1}^M$ be a partition of $\mathcal{V}$. For any $P_j \in \mathcal{P}$, we recursively define $S_0(P_j) = P_j$, and*

$$S_{k+1}(P_j) = \bigcup_{P_{j'} \sim S_k(P_j)} P_{j'}, \quad k = 0, 1, 2, \cdots. \tag{2.27}$$

*$S_k(P_j)$ is called the $k_{\text{th}}$ neighbor patch ball of $P_j$, and $S_k(P_j)/S_{k-1}(P_j)$ the $k_{\text{th}}$ neighbor patch layer of $P_j$.*

**Remark 2.2.13.** *By making use of the notion of neighboring introduced in Definition 2.1.2, 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 2.2.14** (Local approximator)**.** *For each $\psi_i$, let $P_{j_i}$ be the patch such that $\varphi_i \in \Phi_{j_i} \subset \text{span}\{P_{j_i}\}$. Then for each $k \geq 0$, we define the $k$-local approximator of $\psi_i$ as*

$$\begin{aligned} \psi_i^k &= \underset{x \in \text{span}\{S_k(P_{j_i})\}}{\arg\min} \|x\|_A, \\ \text{subject to} \quad \varphi_{i'}^T x &= \delta_{i',i}, \quad \forall\, i' = 1, \ldots, N. \end{aligned} \tag{2.28}$$

**Remark 2.2.15.** *Here $k$ is called the radius of $\psi_i^k$. The condition $\varphi_{i'}^T \psi_i^k = \delta_{i',i}$ is equivalent to $\Phi^T \psi_i^k = \Phi^T \psi_i$. By Lemma 2.2.11 and the definition of $\psi_i^k$, we have*

$$(\psi_i^k - \psi_i)^T A \psi_i = 0, \quad (\psi_i^{k-1} - \psi_i^k)^T A \psi_i^k = 0, \quad \forall k, \tag{2.29}$$

*and hence*

$$\|\psi_i^k\|_A^2 = \|\psi_i\|_A^2 + \|\psi_i^k - \psi_i\|_A^2, \tag{2.30}$$

$$\|\psi_i^{k-1}\|_A^2 = \|\psi_i^k\|_A^2 + \|\psi_i^{k-1} - \psi_i^k\|_A^2. \tag{2.31}$$

**Definition 2.2.16** (Condition factor)**.** *Let $\mathcal{P} = \{P_j\}_{j=1}^M$ be a partition of $\mathcal{V}$, and let $\overline{A}_{P_j}^{-1}$ denote the inverse of $\overline{A}_{P_j}$ as an operator restricted on* $\text{span}\{P_j\}$*. The* ***condition factors*** *are defined as*

$$\delta(P_j, \Phi_j) = \max_{x \in \Phi_j} \frac{x^T x}{x^T \overline{A}_{P_j}^{-1} x}, \quad 1 \leq j \leq M, \quad \text{and} \quad \delta(\mathcal{P}, \Phi) = \max_{P_j \in \mathcal{P}} \delta(P_j, \Phi_j). \tag{2.32}$$

**Remark 2.2.17.**

- *In what follows, since we always fix a choice of $\Phi$ for a partition $\mathcal{P}$, we will simply use $\delta(P_j)$ and $\delta(\mathcal{P})$ to denote $\delta(P_j, \Phi_j)$ and $\delta(\mathcal{P}, \Phi)$ respectively. In particular, when we use the Construction 2.2.6 for $\Phi$ with some integer $q$, we correspondingly use the notations $\delta(\mathcal{P}, q)$.*

- *If we follow the construction $\Phi_j^T\Phi_j = I_{q_j}$, where $q_j$ is the dimension of $\Phi_j$, then we have*

$$\delta(P_j,\Phi_j) = \max_{c\in\mathbb{R}^{q_j}} \frac{c^T\Phi_j^T\Phi_j c}{c^T\Phi_j^T\overline{A}_{P_j}^{-1}\Phi_j c} = \max_{c\in\mathbb{R}^{q_j}} \frac{c^T c}{c^T\Phi_j^T\overline{A}_{P_j}^{-1}\Phi_j c} = \|(\Phi_j^T\overline{A}_{P_j}^{-1}\Phi_j)^{-1}\|_2, \tag{2.33}$$

*that is*

$$(\Phi_j^T\overline{A}_{P_j}^{-1}\Phi_j)^{-1} \preceq \delta(P_j)I_{q_j} \preceq \delta(\mathcal{P})I_{q_j}. \tag{2.34}$$

*Moreover, by block-wise inequalities we have*

$$(\Phi^T(\sum_{j=1}^{M}\overline{A}_{P_j})^{-1}\Phi)^{-1} \preceq \delta(\mathcal{P})I_N. \tag{2.35}$$

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

**Example 2.2.18.** *In this example, we consider the operator A to be the discretization of 2-D second-order elliptic operator by standard 5-point Finite Difference scheme. Similar to the case of graph Laplacian in Example 2.1.10, we have a natural energy decomposition inherited from the assembling of such discretization. Specifically, for every pair of vertices $e_{hori} := [(i,j),(i,j+1)]$ and $e_{vert} := [(i,j),(i+1,j)]$ in the finite difference grid, the energy elements are*

$$E_{hori}^{ij} = -\frac{1}{|e_{hori}|^2}\begin{pmatrix} & (i,j) & & (i,j+1) & \\ (i,j) & 0 & & & \\ & a_{i,j+\frac{1}{2}} & & -a_{i,j+\frac{1}{2}} & \\ & & \ddots & & \\ & -a_{i,j+\frac{1}{2}} & & a_{i,j+\frac{1}{2}} & \\ (i,j+1) & & & & 0 \end{pmatrix}, \text{ and}$$

$$E_{vert}^{ij} = -\frac{1}{|e_{vert}|^2}\begin{pmatrix} & (i,j) & & (i+1,j) & \\ (i,j) & 0 & & & \\ & a_{i+\frac{1}{2},j} & & -a_{i+\frac{1}{2},j} & \\ & & \ddots & & \\ & -a_{i+\frac{1}{2},j} & & a_{i+\frac{1}{2},j} & \\ (i+1,j) & & & & 0 \end{pmatrix}.$$

*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 2.2a shows the high-contrast field (colored in black) in $P_j$. For simplicity, we set $P_j$ as a square domain with fixed length $H = |e_{hori}| = |e_{vert}|$. We also*

*set the coefficient in high-contrast field to be* $10^3$ *(and 1 otherwise). Figure 2.2b 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 purpose. Figure 2.2c shows a similar decreasing trend of the* ***condition factor*** $\delta(P_j, 1)$ *and Figure 2.2d 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 2.2e 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 2.2g plots* $\epsilon(P_j, 1)^2 \cdot \delta(P_j, 1)$ *versus contrast.*

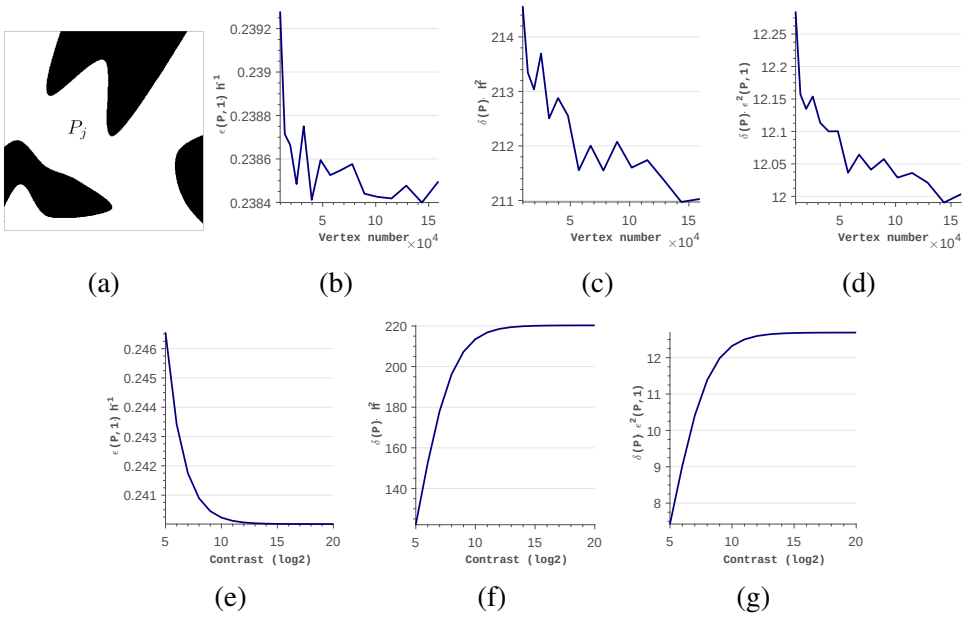



Figure 2.2: An example showing the relationship between mesh size, the **error factor** $\epsilon(P_j, 1)$, the **condition factor** $\delta(P_j, 1)$ and contrast.

The following theorem shows the scaling properties of $\psi_i$, $\psi_i^k$ under Construction 2.2.9 and Definition 2.2.14, which will help to prove the exponential decay of the basis function $\psi_i$.

**Theorem 2.2.19.** *For each* $\psi_i$*, we have*

$$\|\psi_i\|_A \leq \|\psi_i^k\|_A \leq \|\psi_i^0\|_A \leq \sqrt{\delta(P_{j_i})}. \tag{2.36}$$

*Proof.* $\|\psi_i\|_A \le \|\psi_i^k\|_A \le \|\psi_i^0\|_A$ has been proved in the construction of local approximators. We only need to prove $\|\psi_i^0\|_A \le \sqrt{\delta(P_{j_i})}$. Recall that $\psi_i^0$ is defined as

$$\begin{aligned} \psi_i^0 &= \operatorname*{arg\,min}_{x\in\mathrm{span}\{P_{j_i}\}} \|x\|_A, \\ \text{subject to} \quad \varphi_{i'}^T x &= \delta_{i',i}, \quad \forall\, i' = 1,\dots,N. \end{aligned}$$

Without loss of generality, we can assume that $\varphi_i$ is the first column of $\Phi_{j_i}$. And notice that $\|x\|_A = \|x\|_{A_{P_{j_i}}}$, therefore the optimization formation can be rewritten as

$$\begin{aligned} \psi_i^0 &= \operatorname*{arg\,min}_{x\in\mathrm{span}\{P_{j_i}\}} \|x\|_{A_{P_{j_i}}}, \\ \text{subject to} \quad \Phi_{j_i}^T x &= z_i, \end{aligned} \tag{2.37}$$

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

$$\psi_i^0 = A_{P_{j_i}}^{-1}\Phi_{j_i}(\Phi_{j_i}^T A_{P_{j_i}}^{-1}\Phi_{j_i})^{-1} z_i, \tag{2.38}$$

where again $A_{P_{j_i}}^{-1}$ denotes the inverse of $A_{P_{j_i}}$ as an operator restricted to $\mathrm{span}\{P_{j_i}\}$. And thus we have

$$\|\psi_i^0\|_A^2 = \|\psi_i^0\|_{A_{P_{j_i}}}^2 = z_i^T(\Phi_{j_i}^T A_{P_{j_i}}^{-1}\Phi_{j_i})^{-1} z_i. \tag{2.39}$$

Notice that

$$\begin{aligned} \overline{A}_{P_{j_i}} \ge A_{P_{j_i}} \;\Rightarrow\; A_{P_{j_i}}^{-1} \ge \overline{A}_{P_{j_i}}^{-1} \;\Rightarrow\; \Phi_{j_i}^T A_{P_{j_i}}^{-1}\Phi_{j_i} \ge \Phi_{j_i}^T \overline{A}_{P_{j_i}}^{-1}\Phi_{j_i} \\ \Rightarrow\; (\Phi_{j_i}^T \overline{A}_{P_{j_i}}^{-1}\Phi_{j_i})^{-1} \ge (\Phi_{j_i}^T A_{P_{j_i}}^{-1}\Phi_{j_i})^{-1}, \end{aligned}$$

since $\overline{A}_{P_{j_i}}$ and $A_{P_{j_i}}$ are both symmetric, positive definite as operators restricted to $\mathrm{span}\{P_{j_i}\}$. Therefore by (2.35) we have

$$\|\psi_i^0\|_A^2 \le z_i^T(\Phi_{j_i}^T \overline{A}_{P_{j_i}}^{-1}\Phi_{j_i})^{-1} z_i \le \|(\Phi_{j_i}^T \overline{A}_{P_{j_i}}^{-1}\Phi_{j_i})^{-1}\|_2 \|z_i\|_2^2 = \delta(P_{j_i}). \tag{2.40}$$

□

**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 $\mathrm{span}\{P_j\}$, or an orthonormal basis matrix of $U_j$. Namely we have $U_j \subset \mathrm{span}\{P_j\}$, and $\Phi_j^T U_j = \mathbf{0}$. Then we define

$$\alpha(P_j) = \max_{x\in U_j} \frac{x^T \overline{A}_{P_j} x}{x^T \underline{A}_{P_j} x}, \quad 1 \le j \le M, \qquad \text{and} \qquad \alpha(\mathcal{P}) = \max_{P_j\in\mathcal{P}} \alpha(P_j). \tag{2.41}$$

**Remark 2.2.20.**

- *If we choose $\Phi_j$ so that it satisfies condition* (2.17)*, then we have*

$$x^T \underline{A}_{P_j} x = \|x\|^2_{\underline{A}_{P_j}} \geq \frac{1}{\epsilon^2}\|x - P_{\Phi_j} x\|_2^2 = \frac{1}{\epsilon^2}\|x\|_2^2, \quad \forall x \in U_j,$$

*and $x^T \overline{A}_{P_j} x \leq \|\overline{A}_{P_j}\|_2 \|x\|_2^2$. Thus $\alpha(P_j) \leq \epsilon^2 \|\overline{A}_{P_j}\|_2$. This argument is meant to show that we can have $\alpha(\mathcal{P}) < +\infty$ if we choose $\mathcal{P}$ and $\Phi$ properly. But this bound is not tight, as $\alpha(\mathcal{P})$ can be much smaller in general.*

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

$$U_j^T \overline{A}_{P_j} U_j \preceq \alpha(P_j) U_j^T \underline{A}_{P_j} U_j, \quad \forall\, j.$$

The following theorem shows that the local basis function $\tilde{\psi}_i^k$ is exponentially convergent to $\psi_i$ as its support $S_k(P_{j_i})$ extends(or as $k$ increases). Indeed, the exponential decay of $\psi_i$ has been proved in [53, 75, 88, 89] 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 [89] using matrix framework coherent to our energy settings.

**Theorem 2.2.21** (Exponential decay)**.** *For each $\psi_i$, we have*

$$\|\psi_i^k - \psi_i\|_A^2 \leq \Big(\frac{\alpha(\mathcal{P}) - 1}{\alpha(\mathcal{P})}\Big)^k \|\psi_i^0 - \psi_i\|_A^2 \leq \Big(\frac{\alpha(\mathcal{P}) - 1}{\alpha(\mathcal{P})}\Big)^k \delta(P_{j_i}). \tag{2.42}$$

*Proof.* For simplicity, we will write $\psi_i$ as $\psi$, $\psi_i^k$ as $\psi^k$, $P_{j_i}$ as $P$, and $S_k(P_{j_i})$ 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 $P_j \subset \mathcal{V}\backslash S_k$. 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

$$r^k = \psi^k - \psi, \quad k \geq 0; \qquad w^k = \psi^{k-1} - \psi^k, \quad k \geq 1, \tag{2.43}$$

then according to (2.31) we have

$$\|r^{k-1}\|_A^2 = \|r^k\|_A^2 + \|w^k\|_A^2. \tag{2.44}$$

Since $\Phi^T(\psi^k - \psi) = \Phi^T(\psi^{k-1} - \psi^k) = \mathbf{0}$, 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

$$r^{k-1} = P_U^A \psi^{k-1} = U(U^T AU)^{-1} U^T A\psi^{k-1}, \tag{2.45}$$

$$w^k = P_{Y_k}^A \psi^{k-1} = Y_k (Y_k^T A Y_k)^{-1} Y_k^T A\psi^{k-1} = Y_k (Y_k^T A_{S_k} Y_k)^{-1} Y_k^T A_{S_k} \psi^{k-1}. \tag{2.46}$$

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

$$\begin{aligned} \|r^{k-1}\|_A^2 &= \psi^{k-1,T} AU(U^T AU)^{-1} U^T A\psi^{k-1} \\ &= \psi^{k-1,T} A \begin{bmatrix} Y_k & 0 \end{bmatrix} (U^T AU)^{-1} \begin{bmatrix} Y_k^T \\ 0 \end{bmatrix} A\psi^{k-1}. \end{aligned}$$

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

$$U^T AU \succeq U^T \Big( \sum_{j=1}^M \underline{A}_{P_j} \Big) U = \sum_{j=1}^M \begin{pmatrix} 0 & & \\ & U_j^T \underline{A}_{P_j} U_j & \\ & & 0 \end{pmatrix} \succeq \frac{1}{\alpha(\mathcal{P})} \sum_{j=1}^M \begin{pmatrix} 0 & & \\ & U_j^T \overline{A}_{P_j} U_j & \\ & & 0 \end{pmatrix}.$$

As a simple inference of Proposition 2.1.9, we have

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

and therefore

$$\begin{aligned} \sum_{j=1}^M \begin{pmatrix} 0 & & \\ & U_j^T \overline{A}_{P_j} U_j & \\ & & 0 \end{pmatrix} &= \begin{pmatrix} Y_k^T \Big( \sum_{P_j \subset S_k} \overline{A}_{P_j} \Big) Y_k & \\ & Z_k^T \Big( \sum_{P_j \subset \mathcal{V}\backslash S_k} \overline{A}_{P_j} \Big) Z_k \end{pmatrix} \\ &\succeq \begin{pmatrix} Y_k^T A_{S_k} Y_k & \\ & Z_k^T A_{\mathcal{V}\backslash S_k} Z_k \end{pmatrix}. \end{aligned}$$

Combining all results above, we have

$$\begin{aligned} (U^T AU)^{-1} &\preceq \alpha(\mathcal{P}) \begin{pmatrix} Y_k^T A_{S_k} Y_k & \\ & Z_k^T A_{\mathcal{V}\backslash S_k} Z_k \end{pmatrix}^{-1} \\ &= \alpha(\mathcal{P}) \begin{pmatrix} (Y_k^T A_{S_k} Y_k)^{-1} & \\ & (Z_k^T A_{\mathcal{V}\backslash S_k} Z_k)^{-1} \end{pmatrix}, \end{aligned}$$

and thus

$$\begin{aligned} \|r^{k-1}\|_A^2 &\le \alpha(\mathcal{P}) \psi^{k-1,T} A \begin{bmatrix} Y_k & 0 \end{bmatrix} \begin{pmatrix} (Y_k^T A_{S_k} Y_k)^{-1} & \\ & (Z_k^T A_{\mathcal{V}\backslash S_k} Z_k)^{-1} \end{pmatrix} \begin{bmatrix} Y_k^T \\ 0 \end{bmatrix} A\psi^{k-1} \\ &= \alpha(\mathcal{P}) \psi^{k-1,T} A Y_k (Y_k^T A_{S_k} Y_k)^{-1} Y_k^T A\psi^{k-1} \\ &= \alpha(\mathcal{P}) \psi^{k-1,T} A_{S_k} Y_k (Y_k^T A_{S_k} Y_k)^{-1} Y_k^T A_{S_k} \psi^{k-1} \\ &= \alpha(\mathcal{P}) \|w^k\|_A^2. \end{aligned}$$

This gives us

$$\|r^{k-1}\|_A^2 = \|r^k\|_A^2 + \|w^k\|_A^2 \geq \|r^k\|_A^2 + \frac{1}{\alpha(\mathcal{P})}\|r^{k-1}\|_A^2$$
$$\implies \|r^k\|_A^2 \leq \frac{\alpha(\mathcal{P})-1}{\alpha(\mathcal{P})}\|r^{k-1}\|_A^2.$$

Applying this recursively, we have

$$\|\psi^k - \psi\|_A^2 \leq \Big(\frac{\alpha(\mathcal{P})-1}{\alpha(\mathcal{P})}\Big)^k \|\psi^0 - \psi\|_A^2. \tag{2.47}$$

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 2.2.22.**

- *Recall that in Lemma 2.2.8, 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 2.2.21 guides us to choose*

$$k = O(\log\frac{1}{\epsilon} + \log N + \log\delta(\mathcal{P})). \tag{2.48}$$

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

- *By the definition in Equation* (2.41)*, $\alpha(\mathcal{P})$ is locally scaling invariant. Therefore the layer-wise decay rate is unchanged when A is locally multiplied by some scaling constant.*

**Corollary 2.2.23.** *For any $\psi_i$, the interior energy of $\psi_i$ on $\mathcal{V}/S_k(P_{j_i}) = S_k^c(P_{j_i})$ decays exponentially with k, namely*

$$\|\psi_i\|^2_{\underline{A}_{S_k^c(P_{j_i})}} \leq \Big(\frac{\alpha(\mathcal{P})-1}{\alpha(\mathcal{P})}\Big)^k \delta(P_{j_i}).$$

*Moreover, for any two $\psi_i, \psi_{i'}$, we have*

$$|\psi_i^T A \psi_{i'}| \leq \Big(\frac{\alpha(\mathcal{P})-1}{\alpha(\mathcal{P})}\Big)^{\frac{k_{ii'}}{4}-\frac{1}{2}} \delta(\mathcal{P}),$$

*where $k_{ii'}$ is the largest integer such that $P_{j_{i'}} \subset S^c_{k_{ii'}}(P_{j_i})$( or equivalently $P_{j_i} \subset S^c_{k_{ii'}}(P_{j_{i'}})$).*

*Proof.* This proof basically follows the idea in [88]. For any $\psi_i$, recall that $\psi_i^k \in \text{span}(S_k(P_{j_i}))$, thus $\underline{A}_{S_k^c(P_{j_i})}\psi_i^k = \mathbf{0}$, and

$$\|\psi_i\|^2_{\underline{A}_{S_k^c(P_{j_i})}} = \|\psi_i - \psi_i^k\|^2_{\underline{A}_{S_k^c(P_{j_i})}} \leq \|\psi_i - \psi_i^k\|_A^2 \leq \Big(\frac{\alpha(\mathcal{P})-1}{\alpha(\mathcal{P})}\Big)^k \delta(P_{j_i}).$$

For any two $\psi_i, \psi_{i'}$, since $(\psi_i - \psi_i^k)^T \Phi = \mathbf{0}$, and $A\psi_{i'} \in \Phi$, we have $(\psi_i - \psi_i^k)^T A\psi_{i'} = 0$ for all $k$. Also notice that $(\psi_i^k)^T A\psi_{i'}^k = 0$ for $k < \frac{k_{ii'}}{2}$, since $S_k(P_{j_i}) \cap S_k(P_{j_{i'}}) = \emptyset$ when $2k < k_{ii'}$. Therefore taking $k = \lceil \frac{k_{ii'}}{2} \rceil - 1$ we have

$$\begin{aligned} |\psi_i^T A \psi_{i'}| = |(\psi_i^k)^T A(\psi_{i'} - \psi_{i'}^k)| &\leq \|\psi_i^k\|_A \|\psi_{i'} - \psi_{i'}^k\|_A \\ &\leq \Big(\frac{\alpha(\mathcal{P}) - 1}{\alpha(\mathcal{P})}\Big)^{\frac{k}{2}} \delta(\mathcal{P}) \leq \Big(\frac{\alpha(\mathcal{P}) - 1}{\alpha(\mathcal{P})}\Big)^{\frac{k_{ii'}}{4} - \frac{1}{2}} \delta(\mathcal{P}). \end{aligned}$$

□

**Remark 2.2.24.** *Recall that in Lemma 2.2.8 for the compression error with localization $\tilde{\epsilon}_{com}^2$ to be bounded by some prescribed accuracy $\epsilon^2$, we need the localization error $\epsilon_{loc}^2 \leq \frac{\epsilon^2}{N}$. But now with the exponential decaying feature of $\Psi$, empirically, we observe that we can relax the requirement of the localization error to be $\epsilon_{loc}^2 \leq O(\epsilon^2)$ in practice.*

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

**Theorem 2.2.25.** *Let $\lambda_{\min}(A_{st})$ and $\lambda_{\max}(A_{st})$ denote the smallest and largest eigenvalues of $A_{st}$ respectively, then we have*

$$\lambda_{\min}(A_{st}) \geq \lambda_{\min}(A), \qquad \lambda_{\max}(A_{st}) \leq \delta(\mathcal{P}), \tag{2.49}$$

*and so we have*

$$\kappa(A_{st}) = \frac{\lambda_{\max}(A_{st})}{\lambda_{\min}(A_{st})} \leq \delta(\mathcal{P}) \|A^{-1}\|_2. \tag{2.50}$$

*Proof.* Thanks to (2.25), and since $\Phi^T \Phi = I_M$, we have

$$\|A_{\mathrm{st}}^{-1}\|_2 = \|\Phi^T A^{-1} \Phi\|_2 \leq \|A^{-1}\|_2 \quad \Longrightarrow \quad \lambda_{\min}(A_{\mathrm{st}}) \geq \lambda_{\min}(A). \tag{2.51}$$

Thanks to (2.35), and since $A \preceq \sum_{j=1}^M \overline{A}_{P_j}$, we have

$$\delta(\mathcal{P}) I_M \geq (\Phi^T (\sum_{j=1}^M \overline{A}_{P_j})^{-1} \Phi)^{-1} \geq (\Phi^T A^{-1} \Phi)^{-1} = A_{\mathrm{st}} \Longrightarrow \lambda_{\max}(A_{\mathrm{st}}) \leq \delta(\mathcal{P}). \tag{2.52}$$

□

**Corollary 2.2.26.** *Let $\widetilde{\Psi}$ be the local approximator of $\Psi$ defined in Lemma 2.2.8 such that $\|\psi_i - \tilde{\psi}_i\|_A \leq \frac{\epsilon}{\sqrt{N}}$, then we have*

$$\lambda_{\min}(\widetilde{A}_{st}) \geq \lambda_{\min}(A), \qquad \lambda_{\max}(\widetilde{A}_{st}) \leq \left(1 + \frac{\epsilon}{\sqrt{\delta(\mathcal{P})}}\right)^2 \delta(\mathcal{P}), \tag{2.53}$$

*and thus*

$$\kappa(\widetilde{A}_{st}) = \frac{\lambda_{\max}(\widetilde{A}_{st})}{\lambda_{\min}(\widetilde{A}_{st})} \leq \left(1 + \frac{\epsilon}{\sqrt{\delta(\mathcal{P})}}\right)^2 \delta(\mathcal{P})\|A^{-1}\|_2. \tag{2.54}$$

*Proof.* Notice that $P^A_{\widetilde{\Psi}} = \widetilde{\Psi}\widetilde{A}_{\mathrm{st}}^{-1}\widetilde{\Psi}^T A$ is a projection with respect to the energy inner product, we have

$$A\widetilde{\Psi}\widetilde{A}_{\mathrm{st}}^{-1}\widetilde{\Psi}^T A \preceq A \quad \Longrightarrow \quad \widetilde{\Psi}\widetilde{A}_{\mathrm{st}}^{-1}\widetilde{\Psi}^T \preceq A^{-1},$$

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

$$\|\widetilde{A}_{\mathrm{st}}^{-1}\|_2 \leq \|\Phi^T A^{-1}\Phi\|_2 \leq \|A^{-1}\|_2 \quad \Longrightarrow \quad \lambda_{\min}(\widetilde{A}_{\mathrm{st}}) \geq \lambda_{\min}(A).$$

For any $c \in \mathbb{R}^N$, using a similar argument in Lemma 2.2.8, we have

$$\|\Psi c - \widetilde{\Psi} c\|_A \leq C\epsilon\|c\|_2,$$

then we get

$$c^T\widetilde{A}_{\mathrm{st}}c = \|\widetilde{\Psi}c\|_A^2 \leq (\|\Psi c - \widetilde{\Psi}c\|_A + \|\Psi c\|_A)^2 \leq (C\epsilon + \sqrt{\lambda_{\max}(A_{\mathrm{st}})})^2\|c\|_2^2,$$

and using $\lambda_{\max}(A_{\mathrm{st}}) \leq \delta(\mathcal{P})$, we have

$$\lambda_{\max}(\widetilde{A}_{\mathrm{st}}) \leq (1 + \frac{C\epsilon}{\sqrt{\delta(\mathcal{P})}})^2\delta(\mathcal{P}).$$

□

Guided by Theorem 2.2.25, we obtain a simple methodology on the control of condition number of the stiffness matrix $A_{\mathrm{st}}$. As shown in (2.49) and (2.2.25), the only variable is the choice of partition $\mathcal{P}$ and thus the burden again falls to the construction of the partition $\mathcal{P}$. Nevertheless, this new criterion allows us to regulate the quality of partitions directly by avoiding large $\delta(\mathcal{P})$.

Now we can design an algorithm to construct the local approximator $\widetilde{\Psi}$ 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 $\widetilde{\Psi} = \Psi^r$. The localization error can then be guaranteed by Lemma 2.2.8. 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 $P_{j_i}$ by making use of optimization property (2.28). Suppose that we already obtain $\psi_i^{k-1}$, then by optimization property (2.28), one can check that $w_i^k = \psi_i^{k-1} - \psi_i^k$ satisfies the following optimization problem

$$\begin{aligned} w_i^k &= \mathop{\arg\min}_{w \in \mathrm{span}\{S_k(P_{j_i})\}} \|\psi_i^{k-1} - w\|_A, \\ \text{subject to} \quad \Phi^T w &= 0. \end{aligned} \tag{2.55}$$

Similar to the proof of Theorem 2.2.21, let $Y_{i,k}$ denote the joint space of all $U_j$ such that $P_j \subset \{S_k(P_{j_i})\}$. Then the constraints in optimization problem (2.55) imply that $w_i^k \in \mathrm{span}\{Y_{i,k}\}$. Therefore we can explicitly compute $w_i^k$ as

$$w_i^k = P^A_{Y_{i,k}} \psi_i^{k-1} = Y_{i,k} (Y_{i,k}^T A_{S_k(P_{j_i})} Y_{i,k})^{-1} Y_{i,k}^T A_{S_k(P_{j_i})} \psi_i^{k-1}, \tag{2.56}$$

and thus

$$\psi_i^k = \psi_i^{k-1} - Y_{i,k} (Y_{i,k}^T A_{S_k(P_{j_i})} Y_{i,k})^{-1} Y_{i,k}^T A_{S_k(P_{j_i})} \psi_i^{k-1}. \tag{2.57}$$

Specially, we can compute $\psi_i^0$ by (2.57) with an initial guess $\psi_i^{0-1} \in \mathrm{span}\{P_{j_i}\}$ satisfying $\varphi_{i'}^T \psi_i^{0-1} = \delta_{i',i}$, $\forall\ i' = 1, \ldots, N$. 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 $\varepsilon(\mathcal{P}, q)^2 \lambda_{\max}(A) \kappa(Y_{i,k}^T Y_{i,k})$ as we will see in (3.1.1). By choosing all $U_j$ orthonormal, we have $\kappa(Y_{i,k}^T Y_{i,k}) = 1$, then the computation efficiency is measured by $\varepsilon(\mathcal{P}, q)^2 \lambda_{\max}(A)$ if we use CG type method. When we prescribe some certain accuracy $\varepsilon(\mathcal{P}, q)^2$ but $\lambda_{\max}(A)$ is really large, 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 (2.57) with initializer $\psi_i^{k-1}$, and finally ends with $\tilde{\psi}_i = \psi_i^r$ when some stopping criterion is attained for $k = r$. Such inductive computation suggests us to 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

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

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

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

where we have used (2.31). With the analysis above, we propose Algorithm 3 for constructing $\widetilde{\Psi}$.

**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

$$r = O(\log \frac{1}{\epsilon} + \log N + \log \delta(\mathcal{P})). \tag{2.58}$$

Since we are actually compressing $A^{-1}$, we can bound $N$ by the original dimension $n$. Further we assume that locality condition (2.63), (2.64) and (2.65) are satisfied, then the support size of each $\psi_i^r$ is $O(s \cdot r^d)$. Since we also only need to solve (2.57) 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

$$\begin{aligned} &O(\kappa(Y_{i,k}^T A Y_{i,k}) \cdot s \cdot r^d \cdot (\log \frac{1}{\epsilon} + \log N + \log \delta(\mathcal{P}))) \\ &\qquad \leq O(\varepsilon(\mathcal{P}, q)^2 \cdot \lambda_{\max}(A) \cdot s \cdot (\log \frac{1}{\epsilon} + \log n + \log \delta(\mathcal{P}))^{d+1}) \end{aligned} \tag{2.59}$$

Finally the total complexity of Algorithm 3 is $N$ times the cost for every $\psi_i^r$, i.e.

$$O(n \cdot q \cdot \varepsilon(\mathcal{P}, q)^2 \cdot \lambda_{\max}(A) \cdot (\log \frac{1}{\epsilon} + \log n + \log \delta(\mathcal{P}))^{d+1}), \tag{2.60}$$

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

## 2.3 Construction of Partition

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

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

$$\text{subject to} \quad \varepsilon(\widetilde{\mathcal{P}}, q) \leq \epsilon,$$

**Algorithm 3** *Construction of* $\widetilde{\Psi}$

**Input:** Energy decomposition $\mathcal{E}$, partition $\mathcal{P}$, $\Phi$, desired accuracy $\epsilon_{loc}$
**Output:** $\widetilde{\Psi}$

**for** $i = 1, 2, \cdots, \dim(\Phi)$ **do**
  Compute $\psi_i^0$ by solving (2.57) on $\mathcal{S}_0(P_{j_i})$;
  Compute $\psi_i^1$ by solving (2.57) on $\mathcal{S}_1(P_{j_i})$ with initializer $\psi_i^0$;
  **repeat**
    Compute $\psi_i^k$ by solving (2.57) on $\mathcal{S}_k(P_{j_i})$ with initializer $\psi_i^{k-1}$;
    $\eta \leftarrow \frac{\|\psi_i^k - \psi_i^{k-1}\|_A}{\|\psi_i^{k-1} - \psi_i^{k-2}\|_A}$;
  **until** $\frac{\eta^2}{1-\eta^2}\|\psi_i^k - \psi_i^{k-1}\|_A^2 < \epsilon_{loc}^2$;
  $\tilde{\psi}_i \leftarrow \psi_i^k$;
**end for**
$\widetilde{\Psi} = [\tilde{\psi}_0, \tilde{\psi}_1, \ldots, \tilde{\psi}_{\dim(\Phi)}]$.

where $f_1, f_2$ are some penalty functions, $q$ is a chosen integer, and $\epsilon$ 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 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 [19], 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 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 micro-scaled regions.

In view of this, a more reasonable approach is to construct a partition $\mathcal{P}$ based on the information extracted from $A$, which is represented by the local energy decomposition $\mathcal{E}$ 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 2.2: the **error factor** $\epsilon(\mathcal{P}, q)$ and the **condition factor** $\delta(\mathcal{P}, q)$, which keep track of the performance of partition in our

searching approach. These measurements are locally (patch-wisely) 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 $\mathcal{P}$ of $\mathcal{V}$ by iteratively clustering basis functions in $\mathcal{V}$ 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 (2.17)). 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 bound will be explained in Section 3.1. And for large $\delta(P_j, q)$ to diminish, patches with large **condition factor** are combined first. To realize the partitioning procedure and maintain the computation efficiency, we combine patches pair-wisely. Our proposed clustering algorithm is summarized in Algorithm 4 and Algorithm 5.

**Algorithm 4** *Pair-Clustering*

**Input:** energy decomposition $\mathcal{E}$, underlying basis $\mathcal{V}$, desired accuracy $\epsilon$, condition bound $c$.
**Output:** Partition $\mathcal{P}$.

**Initialize:** $P_j = \{v_j\}$, $\delta(P_j, q) = \overline{A}_{P_j}$ (scalar), $1 \le j \le n$;
**while** Number of *active* patches $> 0$, **do**
  Sort *active* $\{P_j\}$ with respect to $\delta(P_j, q)$ in descending order;
  Mark all (*active* and *inactive*) $P_j$ as *unoperated*;
  **for** each active $P_j$ in descending order of $\delta(P_j, q)$, **do**
    **Find_Match**$(P_j, \epsilon, c)$;
    **if Find_Match** succeeds, **then**
      Mark $P_j$ as *operated*;
    **else if** all neighbor patches of $P_j$ are *unoperated*, **then**
      Mark $P_j$ as *inactive*;
    **end if**
  **end for**
**end while**

**Remark 2.3.1.**

- *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* $\varepsilon(P_j \cup P_{j'}, q)^2\delta(P_j \cup P_{j'}, q) \le c$ *implies that the cost is proportional to the gain.*

**Algorithm 5** *Find_Match*

**Input:** $P_j$, $\epsilon$, $c$.
**Output:** Succeeds or Fails.

```
1: for P_{j''} ~ P_j do
2:    Find largest Con(P_j, P_{j''}) among all unoperated P_{j''} (stored as P_{j'});
3: end for
4: Compute ε(P_j ∪ P_{j'}, q) and δ(P_j ∪ P_{j'}, q);
5: if ε(P_j ∪ P_{j'}, q) ≤ ϵ & δ(P_j ∪ P_{j'}, q)ε(P_j ∪ P_{j'}, q)^2 ≤ c, then
6:    combine P_j and P_{j'} to form P_j (P_{j'} no longer exists);
7:    update δ(P_j, q);
8:    return Find_Match succeeds.
9: else
10:   return Find_Match Fails.
11: end if
```

- *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 sub-function **Find_Match**, and can be stored for future use.*

The sub-function **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:

$$\mathrm{Con}(P_j, P_{j'}) = \sum_{E\sim P_j, E\sim P_{j'}} \Big( \sum_{\substack{u\in P_j, v\in P_{j'} \\ u\sim v}} |u^T E v| \Big). \tag{2.61}$$

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

$$\underline{A}_{P_j\cup P_{j'}} = \underline{A}_{P_j} + \underline{A}_{P_{j'}} + \sum_{\substack{E\sim P_j, E\sim P_{j'} \\ E\in P_j\cup P_{j'}}} E = \underline{A}_{P_j} + \underline{A}_{P_{j'}} + \text{Cross Energy}. \tag{2.62}$$

In other words, a larger cross energy implies larger interior eigenvalues of $\underline{A}_{P_j\cup P_{j'}}$, which means $\underline{A}_{P_j\cup P_{j'}}$ 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 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 (2.2.12), we define $\mathcal{N}_1(v) = \mathcal{N}(v)$, and

$$\mathcal{N}_{k+1}(v) = \mathcal{N}(\mathcal{N}_k(v)) = \{u \in \mathcal{V} : u \sim \mathcal{N}_k(v)\},$$

that is, for any $u \in \mathcal{N}_k(v)$, there is a path of length $k$ that connects $u$ and $v$ with respect to the connection relation "$\sim$" defined in Definition 2.1.2. The following definition describes the local interaction property of $A$:

**Definition 2.3.2** (Locality/Sparsity of $A$)**.** *$A$ is said to be local of dimension $d$ with respect to $\mathcal{V}$, if*

$$\#\mathcal{N}_k(v) = O(k^d), \quad \forall k \geq 1, \quad \forall v \in \mathcal{V}. \tag{2.63}$$

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 2.3.3** (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*

$$\varepsilon(\mathcal{N}_k(v), q) = O((hk)^p), \quad \forall k \geq 1, \quad \forall v \in \mathcal{V}. \tag{2.64}$$

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

$$\varepsilon(\mathcal{N}_k(v), q)^2 \delta(\mathcal{N}_k(v), q) \leq c, \quad \forall k \geq 1, \quad \forall v \in \mathcal{V}. \tag{2.65}$$

**Remark 2.3.4.**

- *The locality of $A$ implies that $\#\mathcal{N}_{k+1}(v) - \#\mathcal{N}_k(v) = O(d \cdot k^{d-1})$. In particular $\#\mathcal{N}_1(v) = O(d)$, and thus the number of nonzero entries of $A$ is $m = O(d \cdot n)$.*

- *Let $\mathcal{P}$ be a partition of $\mathcal{V}$ such that each patch $P \in \mathcal{P}$ satisfies* $\operatorname{diam}(P) = O(r)$ *and* $\#P = O(r^d)$*, where $r$ is an integer, and "*diam*" is the path diameter with respect to the adjacency relation "$\sim$" defined in Definition 2.1.2. Let $S_k(P)$ be the layers of neighbors (patch layers) defined in* (2.2.12)*, and $\#^P S_k(P)$ denote the number of patches in $S_k(P)$. Then the locality of $A$ implies that*

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

*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:

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

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 $\mathcal{L}$ and the underlying dimension of $\Omega$ ensure that $A$ is local of dimension $d$ with respect to $\mathcal{V}$. With a consistent discretization of $\mathcal{L}$ 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 [53] prove that if $q \ge \binom{p+d-1}{d}$, then $\varepsilon(\mathcal{N}_k(v), q) \lesssim (hk)^p$ (generalized Poincaré inequality) where $h$ is the fine mesh scale and $p$ is half the order of the elliptic equation; and $\varepsilon(\mathcal{N}_k(v), q)^2 \delta(\mathcal{N}_k(v), q) \le c$ (inverse estimate) for some scaling-invariant constant $c$. Unfortunately these arguments would be compromised if strong ellipticity is not assumed, especially when high-contrast coefficients are present. However, as we see in Example 2.2.18, $\varepsilon(\mathcal{N}_k(v), q)$ and $\delta(\mathcal{N}_k(v), q)$ 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

$$\#\mathcal{P} = O\Big(\frac{n}{(\epsilon^{1/p}/h)^d}\Big) = O\Big(\frac{nh^d}{\epsilon^{d/p}}\Big), \quad \delta(\mathcal{P}, q) = O(\frac{c}{\epsilon^2}), \tag{2.66}$$

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 $P^A_{\widetilde{\Psi}} A^{-1} = \widetilde{\Psi}\widetilde{A}^{-1}_{st}\widetilde{\Psi}^T$ can also take advantage of such locality. In fact, the localization of $\Psi$ not only ensures the efficiency of the construction of $\widetilde{\Psi}$, but also conveys the locality of $A$ to the stiffness matrix $\widetilde{A}_{st}$. Suppose $A$ is local of dimension $d$. Let $\widetilde{\Psi}$ be the local approximator obtained in Algorithm 3 such that $\widetilde{\psi}_i = \psi_i^r,\ 1 \le i \le N$ for some uniform radius $r$. Let $\widetilde{\mathcal{V}} = \{\tilde{v}_i\}_{i=1}^N$ be the orthonormal basis of $\mathbb{R}^N$ such that

$$\tilde{v}_i^T \widetilde{A}_{\mathrm{st}} \tilde{v}_j = \tilde{v}_i^T \widetilde{\Psi}^T A \widetilde{\Psi} \tilde{v}_j = \widetilde{\psi}_i^T A \widetilde{\psi}_j, \quad \forall\ 1 \le i, j \le N.$$

We then similarly define the adjacency relation between basis vectors in $\widetilde{\mathcal{V}}$ with respect to $\widetilde{A}_{\mathrm{st}}$. Since each localized basis $\tilde{\psi}_i$ interacts with patches in $r$ patch layers (thus interacts with other $\tilde{\psi}_{i'}$ corresponding to patches in $2r$ layers), and each patch corresponds to $q$ localized basis functions, using the result in Remark 2.3.4 we have

$$\#\mathcal{N}_k(\tilde{v}) = O((rk)^d \cdot q) = q \cdot r^d \cdot O(k^d), \quad \forall k \ge 1, \quad \forall \tilde{v} \in \widetilde{\mathcal{V}}.$$

That means $\widetilde{A}_{\mathrm{st}}$ inherits the locality of dimension $d$ from $A$. In addition, by using the same argument as in Remark 2.3.4, we have $\#\mathcal{N}_1(\tilde{v}) = O(d \cdot q \cdot r^d)$, which is the number nonzero entries (NNZ) of one single column of $\widetilde{A}_{\mathrm{st}}$. Therefore the number of nonzero entries(NNZ) of $\widetilde{A}_{\mathrm{st}}$ 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 $\epsilon_{loc} = \frac{\epsilon}{\sqrt{N}}$, then the radius has estimate $r = O(\log \frac{1}{\epsilon} + \log n + \log \delta(\mathcal{P}, q))$, and thus the bound on the NNZ of $\widetilde{A}_{\mathrm{st}}$ becomes

$$O(m \cdot q^2 \cdot \frac{1}{s} \cdot (\log \frac{1}{\epsilon} + \log n + \log \delta(\mathcal{P}, q))^d). \tag{2.67}$$

**Choice of $q$**

Recall that, instead of choosing a larger enough $q_j$ for each patch $P_j$ to satisfied $\varepsilon(P_j, q_j) \leq \epsilon$, 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 $2p$ with FEM, we should at least choose $q = \binom{p+d-1}{d}$ 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 $\mathcal{P}$ have the same patch size $s$. Under locality assumption of $A$ given in (2.63), the local operation cost of **Find_Match**$(P_j)$ is approximately

$$O(d \cdot F(\text{size}(P_j))) \leq O(d \cdot F(s)).$$

Here $F(\#P_j)$ is a function of $\#P_j$ that depends only on complexity of solving local eigen problems(with respect to $\underline{A}_{P_j}$) and local inverse problems(with respect to $\overline{A}_{P_j}$) on patch $P_j$, thus we can bound $F(s)$ by $O(s^3)$. Since patches are combined pairwise, 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

$$O(d \cdot \frac{F(s)}{s} \cdot n) + O(n \cdot \log n) = O(d \cdot s^2 \cdot n) + O(n \cdot \log n),$$

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

$$O(d \cdot s^2 \cdot \log s \cdot n) + O(\log s \cdot n \cdot \log n). \tag{2.68}$$

**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

$$O(d \cdot s^2 \cdot \log s \cdot n) + O(\log s \cdot n \cdot \log n) + O\left(q \cdot n \cdot \epsilon^2 \cdot \|A\|_2 \cdot (\log \frac{1}{\epsilon} + \log n + \log \delta(\mathcal{P}))^{d+1}\right), \tag{2.69}$$

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 $\varepsilon(\mathcal{P}, q) \leq \epsilon$.

**Remark 2.3.5.**

- *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* (2.63) *and* (2.64) *are assumed, one can see that* $s \sim (\epsilon^{\frac{1}{p}}/h)^d$*, 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* $\varepsilon$ *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 3.1.*

- *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* $\epsilon^2$*.*

## 2.4 Numerical Examples

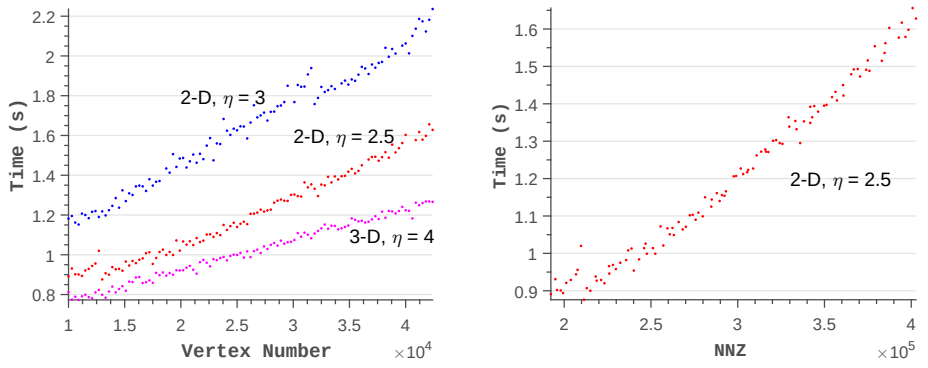
In this section, two numerical examples are reported to demonstrate the efficacy and effectiveness of our proposed operator compression algorithm. For consistency, all the experiments are performed on a single machine equipped with Intel(R) Core(TM) i5-4460 CPU with 3.2GHz and 8GB DDR3 1600MHz RAM.

### 2.4.1 Numerical Example 1: A Graph Laplacian Example

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 $\boldsymbol{x}_i = (x_1^i, \cdots, x_d^i) \in V \subset \mathbb{R}^d$ are generated subject to a uniform distribution over the domain $\Omega = [0, 1]^d$. Edge weights are then given by

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

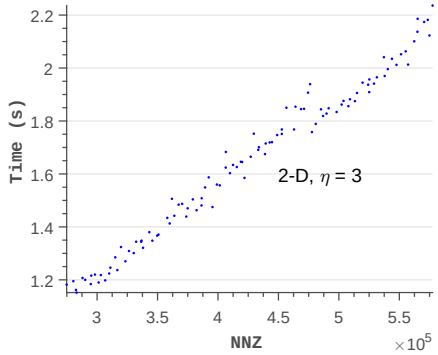
where $r_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2$, $n = \#V$ is the number of vertices and $\eta > 0$ is some density factor for truncating long distance interactions. We set $c_i = 1\ \forall i$ for the sake of well-posedness and invertibility of the graph Laplacian, which gives $\|L^{-1}\|_2 = 1$.



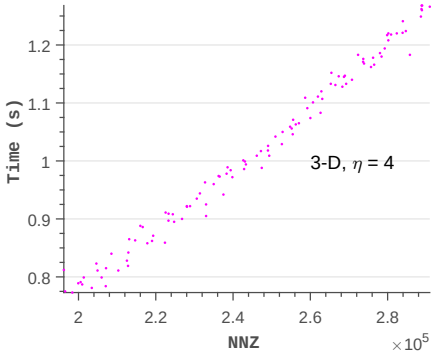

(a) Running time




(b) 2-D graphs with $\eta = 2.5$




(c) 2-D graphs with $\eta = 3$



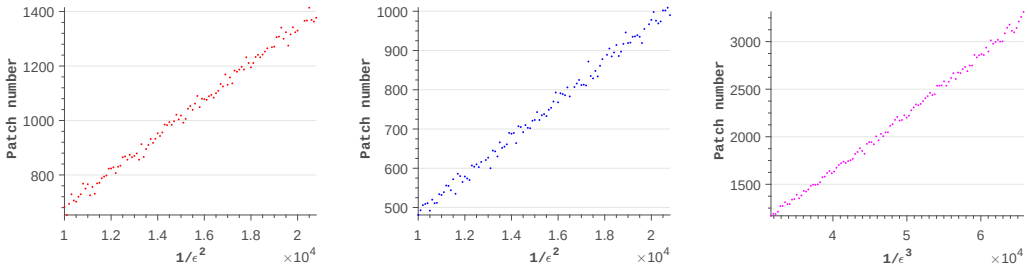

(d) 3-D graphs with $\eta = 4$

Figure 2.3: An 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 nonzero entries (NNZ) in the graph Laplacian operators.

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 $n \propto$ Number of nonzero entries (NNZ) 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 2.1.10, 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 $\frac{1}{\varepsilon^d} \propto n$ 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 2.3 illustrates the nearly-linear time complexity of our algorithm with respect to the graphs' vertex number $n$, which is consistent to our complexity estimation in Section 2.3. Every dot

represents the partitioning result of the given 2-D/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, Figure 2.3b - Figure 2.3d show, respectively, the time complexity of Algorithm 4 versus the NNZ of $L$. Notice that for graphs having NNZ with order up to $10^5$, the running time is still within seconds. These demonstrate the lightweight nature of Algorithm 4.




(a) 2-D graphs: $\#\mathcal{P} \propto \frac{1}{\epsilon^2}$ (b) 2-D graphs: $\#\mathcal{P} \propto \frac{1}{\epsilon^2}$ (c) 3-D graphs: $\#\mathcal{P} \propto \frac{1}{\epsilon^3}$

Figure 2.4: Intrinsic dimension of the sets of graphs.

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 $n \propto \frac{1}{h^d}$ and thus $\#\mathcal{P} \propto \frac{1}{\epsilon^d}$ by the observation in (2.66). The relationship between $\frac{1}{\epsilon^d}$ and patch number $\#\mathcal{P}$ for the three cases is plotted in Figure 2.4a - Figure 2.4c respectively. Figure 2.4a and (2.4b) show the linear relationship between $\#\mathcal{P}$ and $\frac{1}{\epsilon^2}$, meaning that $d = 2$ in these cases. Similarly, the plot in Figure 2.4c that discloses the dimension of the input graphs is 3-dimensional as $\#\mathcal{P} \propto \frac{1}{\epsilon^3}$. 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 $\boldsymbol{x}_i$, we denote $\mathrm{NN}(\boldsymbol{x}_i, k_i)$ to be the set of the $k_i$ nearest vertices of $\boldsymbol{x}_i$. Any two vertices $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ have an edge of weight $w_{ij} = 1/r_{ij}^2$ if and only if $\boldsymbol{x}_i \in \mathrm{NN}(\boldsymbol{x}_j; k_j)$ or $\boldsymbol{x}_j \in \mathrm{NN}(\boldsymbol{x}_i; k_i)$. Let $\boldsymbol{y} = (0.5, 0.5)$ be the center of $\Omega$. We set $k_i = 15$ if $\|\boldsymbol{x}_i - \boldsymbol{y}\|_2 \leq 0.25$ and $k_i = 5$ otherwise. Therefore the sub-graph inside the disk $B(\boldsymbol{y}, 0.25)$ has a stronger connectivity than the sub-graph 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 2.5a shows the ratios $\epsilon^2_{\text{com}}/\epsilon^2$ and $\varepsilon(\mathcal{P}, 1)^2/\epsilon^2$, where $\epsilon^2_{\text{com}} = \|L^{-1} - P^L_\Psi L^{-1}\|_2$ is the compression error. Using Algorithm 4, we achieve a nearly optimal local error control. Also notice that the global compression error ratio $\epsilon^2_{\text{com}}/\epsilon^2$ is strictly bounded by 1 but also above 0.5, meaning that our approach is neither playing down nor overdoing the compression. Figure 2.5b shows the condition number of $A_{\text{st}}$, which is consistent to the prescribed accuracy, and is strictly bounded by $\epsilon^2 \cdot \delta(\mathcal{P}, 1)$ and the prescribed condition bound $c$. Figure 2.5c plots patch number $\#\mathcal{P}$ 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 Equation (2.63). Therefore the curve is below linear, which is consistent to estimate (2.66) with $d = 2$. As comparison, the red curve is the optimal compression dimension subject to the same prescribed accuracy given by eigenvalue decomposition. Since Algorithm 4 combines patches pair-wisely, the output patch number $\#\mathcal{P}$ can be up to 2 times the optimal case. Figure 2.5d shows the partition result with $\#\mathcal{P} = 298$ for the case $\epsilon^2 = 0.001$, where the black lines outline the boundaries of patches. Figure 2.5e 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 $\epsilon^2 = 0.0001$ to study the performance of compression with localization. In this case we have $N = \#\mathcal{P} = \dim(\Phi) = 1446$. Let $\widetilde{\Psi}$ be the local approximator of $\Psi$ constructed by Algorithm 3 subject to localization error $\|\tilde{\psi}_i - \psi_i\|^2_A \le \epsilon^2_{loc}$. Figure 2.6 shows the compression error $\tilde{\epsilon}^2_{\text{com}} = \|L^{-1} - P^L_{\widetilde{\Psi}} L^{-1}\|_2$, mean radius and mean support size of $\widetilde{\Psi}$ with different $\epsilon^2_{loc}$ varies from 0.1 to 0.0001. Recall that Lemma 2.2.8 requires a localization error $\epsilon^2_{loc} = \epsilon^2/N$ to ensure $\tilde{\epsilon}^2_{\text{com}} \le \epsilon^2$, but Figure 2.6a shows that $\epsilon^2_{loc} = \epsilon^2$ is adequate. Figure 2.6b shows the linearity between mean radius and $\log \frac{1}{\epsilon_{loc}}$, which is consistent to the exponential decay of $\Psi$ proved in Theorem 2.2.21. Figure 2.6c shows the quadratic relation between mean support size and mean radius of $\widetilde{\Psi}$, which again reflects the geometric dimension of the graph is 2.

By fixing $\epsilon^2_{loc} = 0.0001$, we have the mean radius of $\widetilde{\Psi} \approx 4.5$ 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 2.7a-2.7f (first two rows of Figure 2.7) show the profiles of $|\psi_i|$ and

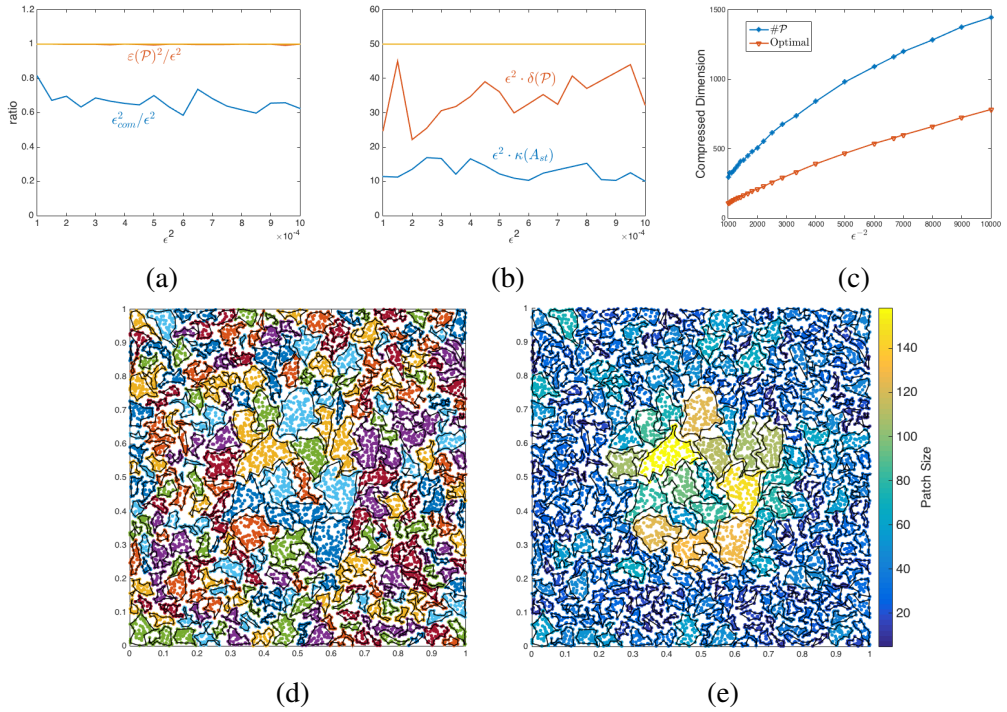
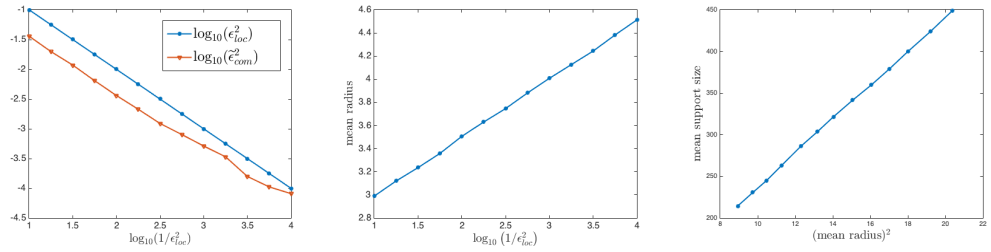



Figure 2.5: Error and well-posedness studies of the compressed operators.

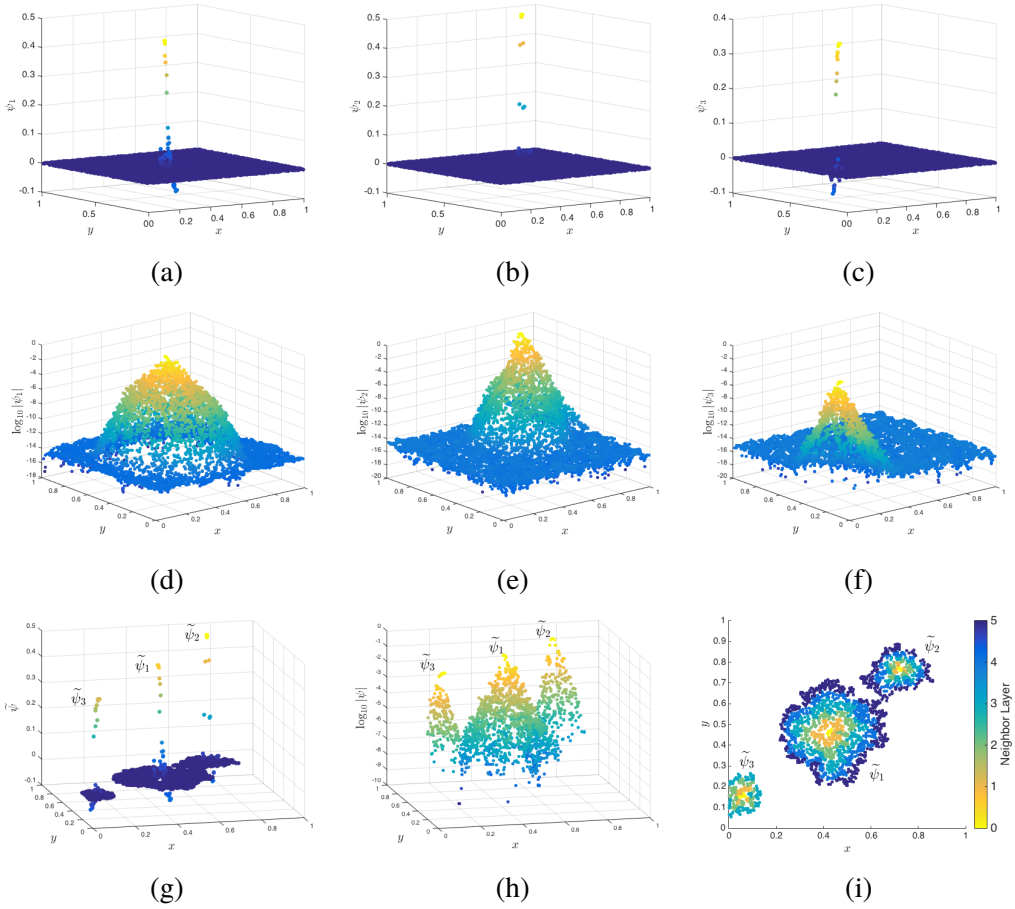
$\log_{10}|\psi_i|$ for $i = 1, 2, 3$. Though all of them decay exponentially from their center patches to outer layers, $\psi_1$ decays slower than $\psi_3$ since the graph has a higher connectivity (i.e., larger patch sizes) near the center. Figure 2.7g and Figure 2.7h show the profiles of $|\tilde{\psi}_i|$ and $\log_{10}|\tilde{\psi}_i|$ for $i = 1, 2, 3$. The bird's-eye view of their supports is shown in Figure 2.7i. Similarly, $\tilde{\psi}_1$ needs a larger support than $\tilde{\psi}_3$ to achieve the same accuracy, which implies that $\psi_1$ decays slower than $\psi_3$. Figure 2.8 shows the spectrum of $L^{-1}$, $L^{-1} - P^L_\Psi L^{-1}$ and $L^{-1} - P^L_{\widetilde{\Psi}} L^{-1}$. Notice that if we truncate the fine-scale part of $L^{-1}$ with prescribed accuracy $\epsilon^2$, then $L^{-1} - P^L_\Psi L^{-1}$ has rank $n - N$ and the compression error is $8.13 \times 10^{-5} < \epsilon^2$. Similarly, if the local approximator $\widetilde{\psi}$ is applied (instead of $\psi$), then $L^{-1} - P^L_{\widetilde{\Psi}} L^{-1}$ also has rank $n - N$, and the compression error is also $8.13 \times 10^{-5} < \epsilon^2$. This means that the same compression error is achieved by the compression $P^L_{\widetilde{\Psi}} L^{-1}$ with localization.

Fourth, we compare our results with the compression given by the PCA [55], that is, $L^{-1} \approx \sum_{i=1}^{N_{\text{PCA}}} \lambda_i^{-1} q_i q_i^T$, where $\lambda_i$ is the $i_{\text{th}}$ 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 global eigen problem and the compressed operator $\sum_{i=1}^{N_{pca}} \lambda_i^{-1} q_i q_i^T$




(a) (b) (c)

Figure 2.6: Compression error $\epsilon^2_{\text{com}}$ and the mean radius of $\widetilde{\Psi}$.




(a) (b) (c)

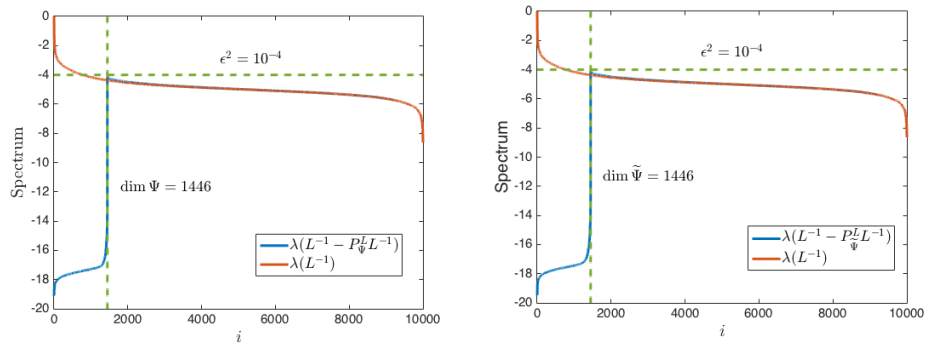
(d) (e) (f)

(g) (h) (i)

Figure 2.7: Profiles of $\psi_1$, $\psi_2$, $\psi_3$, and the corresponding approximator $\tilde{\psi}_1$, $\tilde{\psi}_2$, $\tilde{\psi}_3$.

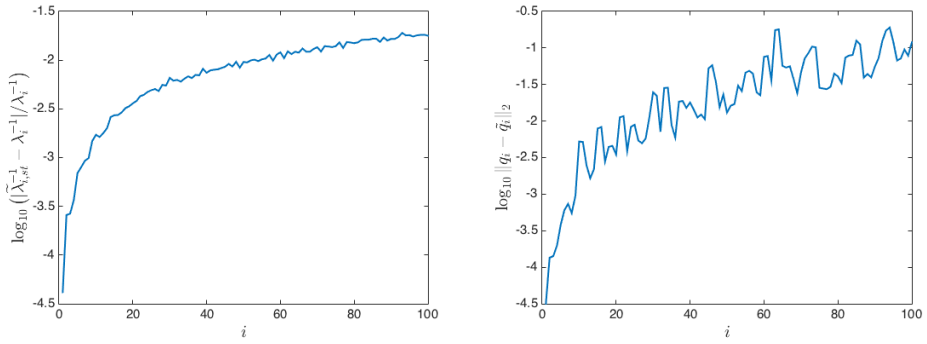
usually loses the original sparsity features. On the other hand, our approach has a larger number of basis functions ($N = 1446$) since only local eigen information is used and patches are combined pair-wisely. But our approach gives local functions with compressed dimension just up to 2 times of the optimal dimension. Further, it turns out that we can recover the eigenvectors of $L$ corresponding to relatively small eigenvalues by solving eigenvalue problem of $A_{\text{st}}$ (or $\widetilde{A}_{\text{st}}$). Figure 2.10 shows the

$2^{\text{nd}}$, $10^{\text{th}}$, $20^{\text{th}}$ and $50^{\text{th}}$ eigenvectors corresponding to the small eigenvalues of $L^{-1}$ (first row) and $\widetilde{A}_{\text{st}}^{-1}$ (second row) respectively. Let $\tilde{\lambda}_{i,st}$ be the $i_{\text{th}}$ smallest eigenvalue of $\widetilde{A}_{\text{st}}$, and $\tilde{\xi}_i$ be the corresponding eigenvector so that $\tilde{q}_i = \widetilde{\Psi}\tilde{\xi}_i$ has $l_2$-norm equal to 1. From the experiment, we observe that $\tilde{\lambda}_{i,st}^{-1}$ is a good approximation of $\lambda_i^{-1}$ and $\tilde{q}_i$ is a good approximation of $q_i$ for small $\lambda_i$, as shown in Figure 2.9. In other words, this procedure provides us convenience for computing the first few eigenvalues and eigenvectors of $L$, since $\widetilde{A}_{\text{st}}$ has the compressed size with a much smaller condition number ($\kappa(\widetilde{A}_{\text{st}}) = 1.14 \times 10^5$ vs. $\kappa(L) = 4.29 \times 10^8$).




(a) Spectrum of $L^{-1}$ and $L^{-1} - P_{\Psi}^{L} L^{-1}$ (b) Spectrum of $L^{-1}$ and $L^{-1} - P_{\widetilde{\Psi}}^{L} L^{-1}$

Figure 2.8: Spectrum of $L^{-1}$, $L^{-1} - P_{\Psi}^{L} L^{-1}$ and $L^{-1} - P_{\widetilde{\Psi}}^{L} L^{-1}$.



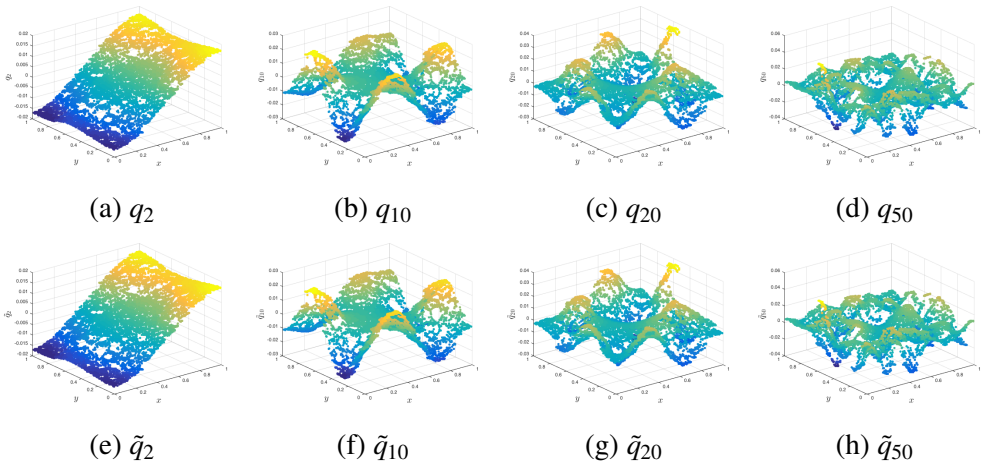

(a) (b)

Figure 2.9: Difference between the true eigenvalues/eigenvectors and the approximated ones.

### 2.4.2 Numerical Example 2: A PDE Example

Our second numerical example arises from using GFEM to solve the following elliptic equation with homogeneous Dirichlet boundary conditions:

$$-\nabla \cdot (a \cdot \nabla u) = f, \quad u \in H_0^1(\Omega), \tag{2.70}$$



(a) $q_2$ (b) $q_{10}$ (c) $q_{20}$ (d) $q_{50}$

(e) $\tilde{q}_2$ (f) $\tilde{q}_{10}$ (g) $\tilde{q}_{20}$ (h) $\tilde{q}_{50}$

Figure 2.10: Plot of 2nd, 10th, 20th and 50th eigenvectors corresponding to $L^{-1}$ (first row) and $\widetilde{A}_{\text{st}}^{-1}$ (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

$$a = \begin{pmatrix} a_{11} & a_{12} \\ a_{12} & a_{22} \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} \mu \cdot e_1 & 0 \\ 0 & \mu \cdot e_2 \end{pmatrix} \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}. \tag{2.71}$$

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 $w_1(x, y), w_2(x, y) \overset{i.i.d}{\sim} \mathcal{U}([-0.1, 0.1])$, a uniform distribution on $[-0.1, 0.1]$. The contrast factor $\mu(x, y)$ is generated from the background permeability as shown in Figure 2.11a. $\theta$ is given by $\theta(x, y) = \pi \cdot (x + y)$. The magnitude of $|a_{11}(x, y)|$ in $\Omega$ is also plotted in Figure 2.11d as a reference.

We use GFEM [19] with a regular triangularization to form a finite system that is fine enough to capture the details of the background field. The basis $\mathcal{V} \in \mathbb{R}^n$ is the vector representation of the Galerkin nodal basis and the PD matrix $A$ is the stiffness matrix of nodal basis with respect to the energy inner product $\int_\Omega (\nabla\cdot)^T a (\nabla\cdot) dxdy$. The energy decomposition $\mathcal{E} = \{E_k\}_{k=1}^m$ is the collection of all patch-wise stiffness

matrices $E_k$ on every triangle $\tau_k$. Specifically, each $E_k$ has the form

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

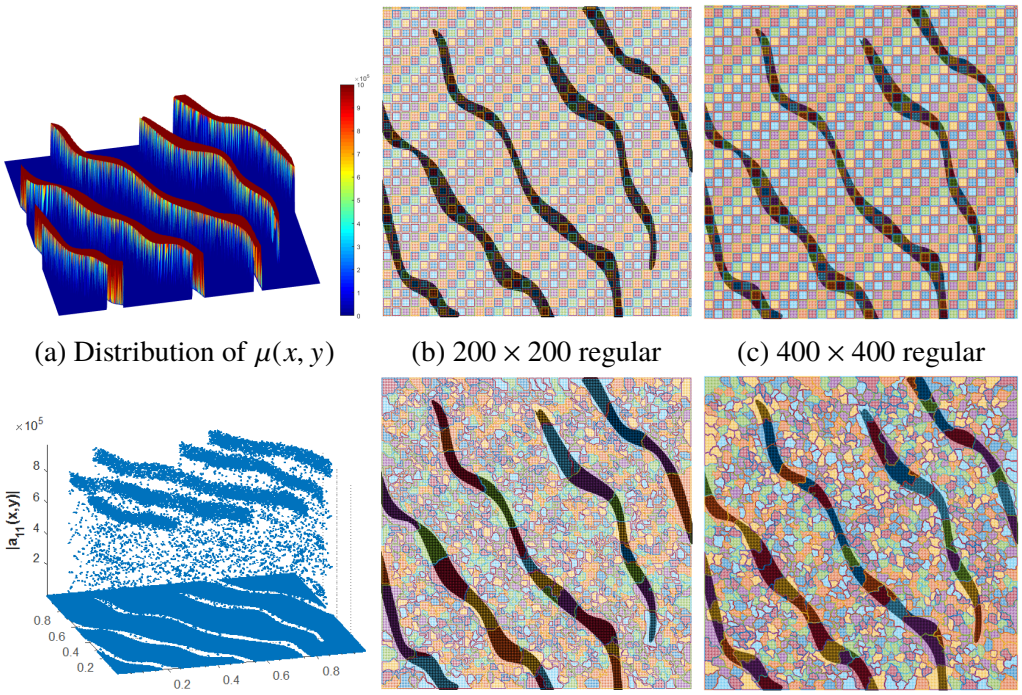
where $\phi_{i_k}$, $i = 1,2,3$ are the three nodal basis surrounding $\tau_k$. In this case, every finest energy element involves three functions, which generalizes the concept of graphs' edges as we mentioned before. One should also notice that for patch $\tau_k$ touching the boundary of $\Omega$, the corresponding $E_k$ reduces to involve only two or one function since nodal basis functions on boundary are not required for homogeneous Dirichlet boundary conditions. Moreover, following the discussion in Section 2.3, we will choose $q = 1$ since the problem (2.70) is of second order. That is, we only construct one measurement function $\varphi$ on each patch of the partition.

| | Partition | $\#\mathcal{P}$ | $\varepsilon(\mathcal{P})^2$ | $\delta(\mathcal{P})$ | $\kappa(A_{\text{st}})$ | $\delta(\mathcal{P})\|A^{-1}\|_2$ | $\max \delta(P_j)\varepsilon(P_j)^2$ |
|---|---|---|---|---|---|---|---|
| $200 \times 200$ | Ours | 1396 | $9.99 \times 10^{-5}$ | $3.87 \times 10^{6}$ | $4.80 \times 10^{3}$ | $1.15 \times 10^{4}$ | 233.31 |
| | Regular | 1156 | $5.77 \times 10^{-5}$ | $2.53 \times 10^{11}$ | $2.53 \times 10^{8}$ | $7.49 \times 10^{8}$ | 454.70 |
| $400 \times 400$ | Ours | 1199 | $9.99 \times 10^{-5}$ | $4.05 \times 10^{6}$ | $3.12 \times 10^{3}$ | $1.11 \times 10^{4}$ | 277.60 |
| | Regular | 1156 | $5.05 \times 10^{-5}$ | $3.14 \times 10^{11}$ | $2.26 \times 10^{8}$ | $8.59 \times 10^{8}$ | $1.06 \times 10^{3}$ |

Table 2.1: Comparison with the uniform regular partitioning for $200 \times 200$ and $400 \times 400$ resolutions.

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 $\epsilon^2 = 10^{-4}$ 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 achieves the prescribed accuracy (i.e., $\varepsilon(\mathcal{P}, 1)^2 < 10^{-4}$). Notice that in this case, the first eigenvector, $\Phi_j$ on every patch is a constant. Such choice of $\Phi$, along with the use of regular partition is equivalent to the set up in [53, 88]. Therefore, the only modification we apply in this numerical example is the adaptive construction of the construction $\mathcal{P}$, 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 2.1 summarizes the partitioning results in both cases. Under regular partitioning, we have $\#\mathcal{P} = 1156$ and each $P_j$ has the size of at most $6 \times 6$ (Figure 2.11b) and




(a) Distribution of $\mu(x, y)$ (b) $200 \times 200$ regular (c) $400 \times 400$ regular

(d) $|a_{11}(x, y)|$ for $(x, y) \in \Omega$ (e) $200 \times 200$ proposed (f) $400 \times 400$ proposed

Figure 2.11: Partitioning result: operator stemmed from the FEM of an elliptic PDE with high-contrast coefficients. (a) shows the distribution of the high-contrast factor in the domain $\Omega$. (b) and (c) shows the regular partition in resolution $200 \times 200$ and $400 \times 400$ respectively. (d) shows the value of $a_{11}(x, y)$ on $\Omega$. (e) and (f) show the partition results obtained from Algorithm 4.

$12 \times 12$ (Figure 2.11c) vertices respectively. Notably, the **condition factors** for both cases go up to $10^{11}$ and the corresponding true condition numbers $\kappa(A_{\text{st}})$ 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** $\varepsilon(\mathcal{P}, 1)^2$ achieved in both cases are strictly bounded above by the prescribed accuracy $\epsilon^2 = 10^{-4}$ and $\delta(\mathcal{P}, 1)$ is of the order of $10^6$ only. Indeed, the true compression error is even smaller as the square **error factor** $\varepsilon(\mathcal{P}, 1)^2$ is only the theoretical upper bound as required in Proposition 2.2.2. Furthermore, the true condition numbers $\kappa(A_{\text{st}})$ are in the order of $10^3$ (compare to $10^8$ from regular partitioning), which are again bounded by (and is much smaller than) $\delta(\mathcal{P}, 1)\|A^{-1}\|_2$ as observed in Theorem 2.2.25. Moreover, the patch number $\#\mathcal{P}$ is comparable to the case of regular partitioning. Also notice that $\max_{P_j \in \mathcal{P}} \delta(P_j, 1) \cdot \varepsilon(P_j, 1)^2 < c = 300$ 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 2.11e and Figure 2.11f

respectively.

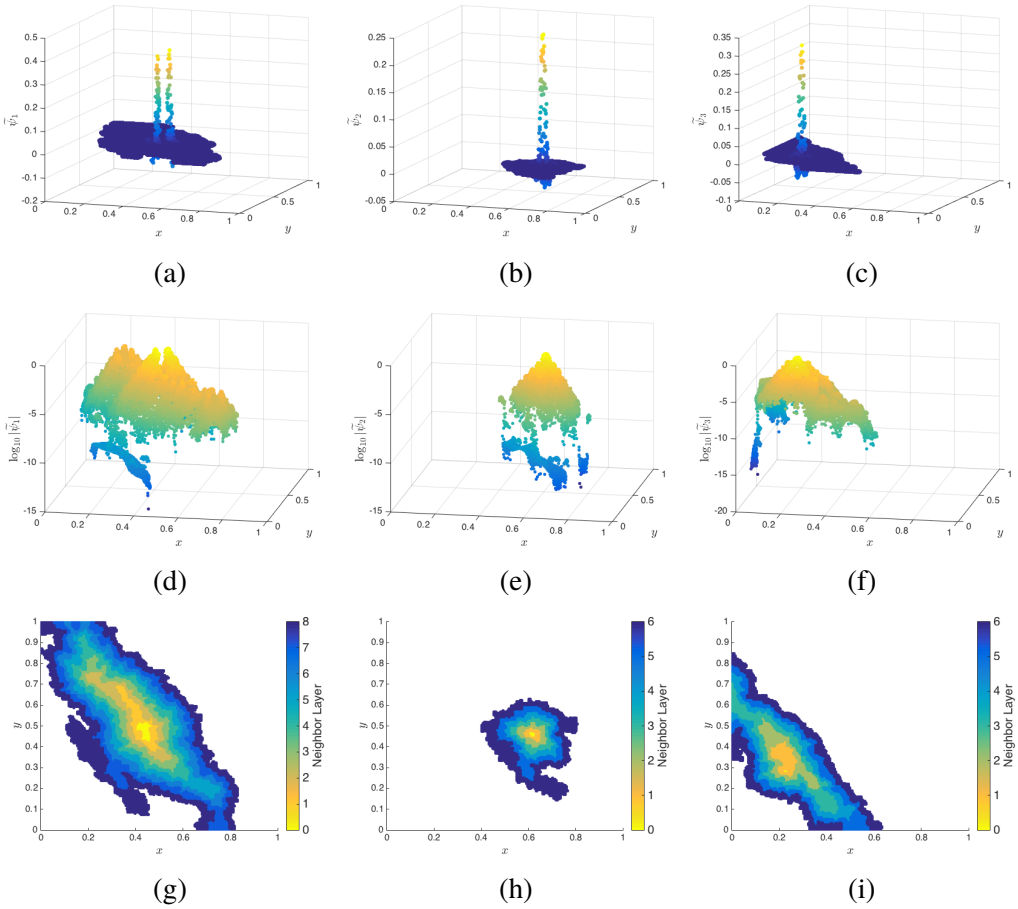



Figure 2.12: Samples of the localized basis functions.

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, corresponding patch-wise **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 avoided. As shown in Figure 2.11f and Figure 2.11e, 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 2.12. We can see

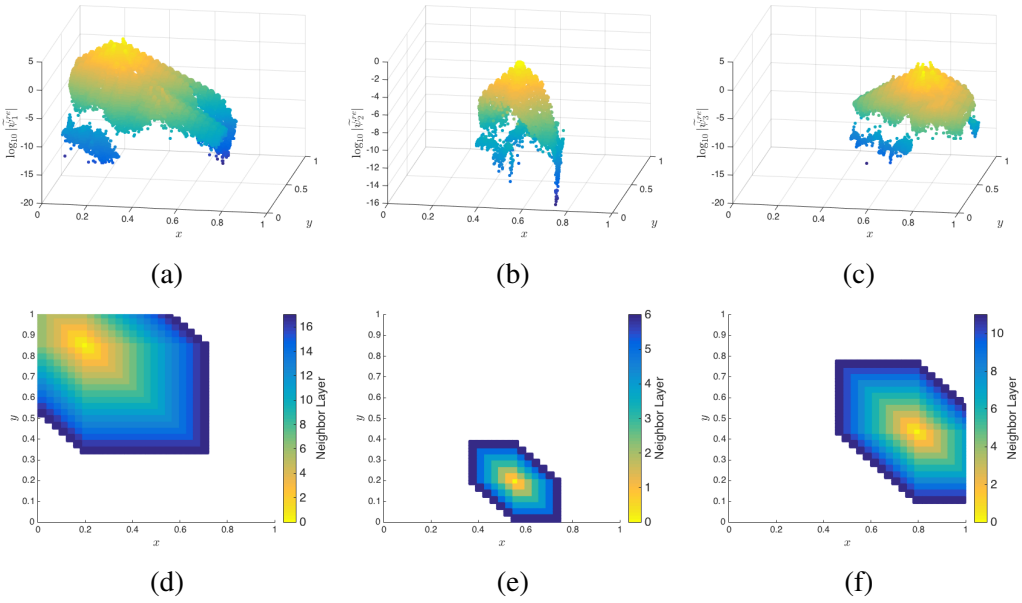



Figure 2.13: Samples of the localized basis functions from a regular partition.

in Figure 2.12a 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 $(\widetilde{\psi}_1, \widetilde{\psi}_3)$ have to extend along the high permeability channels and thus end up with relatively large supports. Recall that Remark 2.2.22 implies that the decay rate of $\psi_i$ 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 2.13 plots the log-profile and the supports of three localized functions $\tilde{\psi}_1^{re}, \tilde{\psi}_2^{re}, \tilde{\psi}_3^{re}$ 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** $\delta(\mathcal{P}, 1)$. Recall that to achieve an desired localization error, the radius of localized basis is also affected by $\log \delta(\mathcal{P}, 1)$.

*Chapter 3*

# MULTIRESOLUTION MATRIX DECOMPOSITION

In this chapter, we extend our preceding operator compression method to a MMD framework. Our MMD method is the product of a hierarchical partitioning procedure and an associated construction of a nested basis structure, which passes the energy decomposition from the finest level to the coarsest level. It resolves the large condition number of a PD matrix by decomposing it into many well-conditioned pieces. This MMD scheme naturally leads to a parallelizable fast linear solver for PD linear systems.

In Section 3.1, we systematically explain the how to generalize our operator compression method to a MMD framework, based on a hierarchically inherited energy decomposition. In Section 3.2, we propose the concept of localization of MMD and the inherited locality of the compressed operator. These essential ingredients guides us to develop the parallelizable solver with nearly-linear time complexity for large and sparse PD linear systems. We further discuss our MMD frame in Section 3.3 from the perspective of a multilevel operator compression. Section 3.4 is dedicated to several graph Laplacian examples that illustrate the effectiveness of our MMD approach in resolving the large condition number of a PD matrix. Error estimate and numerical results are reported to show the efficacy of this proposed algorithm.

## 3.1 Multiresolution Matrix Decomposition (MMD)

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 meanwhile, 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 (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 sub-system to be solved on each level is as sparse as the original $A$. Here we will make use of the choice of the partition $\mathcal{P}$, the basis $\Phi$ and $\Psi$ obtained in Section 2.2 and Section 2.3 to serve the purpose of multiresolution operator decomposition. In the following, We first implement a **one-level decomposition**.

### 3.1.1 One-level Operator Decomposition

Let $\mathcal{P}$, $\Phi$, $\Psi$ and $U$ be constructed as in Algorithm 1, namely

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

(ii) $\Phi = [\Phi_1, \Phi_2, \cdots, \Phi_M]$ 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) $U = [U_1, U_2, \cdots, U_M]$ such that every $U_j \subset \text{span}\{P_j\}$ has dimension $\dim(U_j) = \dim(\text{span}\{P_j\}) - q_j$ and satisfies $\Phi_j^T U_j = \mathbf{0}$.

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

$$U^T A\Psi = U^T\Phi(\Phi^T A^{-1}\Phi)^{-1} = 0. \tag{3.1}$$

Thus the inverse of $A$ can be written as

$$\begin{aligned} A^{-1} &= \Big( \begin{bmatrix} U^T \\ \Psi^T \end{bmatrix}^{-1} \begin{bmatrix} U^T \\ \Psi^T \end{bmatrix} A \begin{bmatrix} U & \Psi \end{bmatrix} \begin{bmatrix} U & \Psi \end{bmatrix}^{-1} \Big)^{-1} \\ &= U(U^T AU)^{-1}U^T + \Psi(\Psi^T A\Psi)^{-1}\Psi^T. \end{aligned} \tag{3.2}$$

In the following we denote $\Psi^T A\Psi = A_{\text{st}}$ and $U^T AU = B_{\text{st}}$ respectively. We also use the phrase "solving $A^{-1}$" to mean "solving $A^{-1}b$ for any $b$". From (3.2), we observe that solving $A^{-1}$ is equivalent to solving $A_{\text{st}}^{-1}$ and $B_{\text{st}}^{-1}$ separately. For $B_{\text{st}}$, notice that since the space/basis $U$ is constructed locally with respect to each patch $P_j$, $B_{\text{st}}$ will inherit the sparsity characteristic from $A$ if $A$ is local/sparse. Thus it will be efficient to solve $B_{\text{st}}^{-1}$ using iterative type methods if the condition number of $B_{\text{st}}$ is bounded. In the following, we introduce Lemma 3.1.1 which provides an upper bounded of the $B_{\text{st}}$ that ensures the efficiency of solving $B_{\text{st}}^{-1}$. The proof of the

lemma imitates the proof from Theorem 10.9 of [89], where the required condition (2.18) corresponds to Equation (2.3) in [89].

**Lemma 3.1.1.** *If* $\Phi$ *satisfies the condition* (2.18) *with constant* $\epsilon$*, then*

$$\lambda_{\max}(B_{st}) \leq \lambda_{\max}(A) \cdot \lambda_{\max}(U^T U), \qquad \lambda_{\min}(B_{st}) \geq \frac{1}{\epsilon^2} \cdot \lambda_{\min}(U^T U), \tag{3.3}$$

*and thus*

$$\kappa(B_{st}) \leq \epsilon^2 \cdot \lambda_{\max}(A) \cdot \kappa(U^T U). \tag{3.4}$$

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

$$\begin{aligned}\lambda_{\max}(B_{\mathrm{st}}) = \|B_{\mathrm{st}}\|_2 = \|U^T A U\|_2 \leq \|A\|_2 \|U\|_2^2 \\ = \|A\|_2 \|U^T U\|_2 = \lambda_{\max}(A)\lambda_{\max}(U^T U).\end{aligned}$$

For $\lambda_{\min}(B_{\mathrm{st}})$, since $\Phi$ satisfies the condition (2.18) with constant $\epsilon$ and $\Phi^T U = \mathbf{0}$, we have

$$\|x\|_2^2 \leq \frac{1}{\lambda_{\min}(U^T U)} x^T U^T U x \leq \frac{\epsilon^2}{\lambda_{\min}(U^T U)} x^T U^T A U x = \frac{\epsilon^2}{\lambda_{\min}(U^T U)} x^T B_{\mathrm{st}} x,$$

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

We shall have some discussions 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

$$\kappa(U^T U) = \frac{\lambda_{\max}(U^T U)}{\lambda_{\min}(U^T U)} = \frac{\max_{1 \leq j \leq M} \lambda_{\max}(U_j^T U_j)}{\min_{1 \leq j \leq M} \lambda_{\min}(U_j^T U_j)}. \tag{3.5}$$

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 [102]. 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 $\mathcal{P}$ and an integer $q$ so that $\Phi$ satisfies condition (2.18) with constant $\varepsilon(\mathcal{P}, q)$, thus the posterior bound of $\kappa(B_{\mathrm{st}})$ is $\varepsilon(\mathcal{P}, q)^2 \lambda_{\max}(A)$(when $\kappa(U^T U) = 1$). Recall that $\kappa(A_{\mathrm{st}})$ is bounded by $\delta(\mathcal{P}, q)\|A^{-1}\|_2$, therefore $\kappa(A_{\mathrm{st}})\kappa(B_{\mathrm{st}}) \leq \varepsilon(\mathcal{P}, q)^2 \delta(\mathcal{P}, q)\kappa(A)$. That is, $A_{\mathrm{st}}$ and $B_{\mathrm{st}}$ divide the burden of the large condition number of $A$ with an amplification

factor $\varepsilon(\mathcal{P},q)^2\delta(\mathcal{P},q)$. We call $\kappa(\mathcal{P},q) \triangleq \varepsilon(\mathcal{P},q)^2\delta(\mathcal{P},q)$ the $q_{\text{th}}$-order condition number of the partition $\mathcal{P}$. This explains why we attempt to bound $\kappa(\mathcal{P},q)$ in the construction of the partition $\mathcal{P}$.

Ideally, we hope the one-level operator decomposition gives $\kappa(A_{\text{st}}) \approx \kappa(B_{\text{st}})$, so that the two parts equally share the burden in parallel. But such result may not be good enough when $\kappa(A_{\text{st}})$ and $\kappa(B_{\text{st}})$ 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 $\kappa(B_{\text{st}})$; then if $\kappa(A_{\text{st}})$ is still large, we apply the decomposition to $A_{\text{st}}^{-1}$ again to further decompose $\kappa(A_{\text{st}})$. However, the decomposition of $A^{-1}$ is based on the construction of $\mathcal{P}$ and $\Phi$, namely on the underlying energy decomposition $\mathcal{E} = \{E_k\}_{k=1}^m$ of $A$. Hence, we have to construct the corresponding **energy decomposition** of $A_{\text{st}}$ before we implement the same operator decomposition on $A_{\text{st}}^{-1}$.

### 3.1.2 Inherited Energy Decomposition

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

$$E_k^\Psi = \Psi^T E_k \Psi, \quad k = 1, 2, \cdots, m. \tag{3.6}$$

Notice that this inherited energy decomposition of $A_{\text{st}}$ with respect to $\mathcal{E}$ has the same number of energy elements as $\mathcal{E}$, which is not preferred and actually redundant in practice. Therefore we shall consider reducing the energy decomposition of $A_{\text{st}}$. Indeed we will use $\widetilde{\Psi}$ instead of $\Psi$ in practice, where each $\tilde{\psi}_i$ is some local approximator of $\psi_i$ (obtained by Construction 2.2.9). Specifically, we will actually deal with $\widetilde{A}_{\text{st}} = \widetilde{\Psi}^T A\widetilde{\Psi}$ and thus we shall consider to find a proper condensed energy decomposition of $\widetilde{A}_{\text{st}}$.

If we see $\widetilde{A}_{\text{st}}$ as a matrix with respect to the reduced space $\mathbb{R}^N$, then for any vector $\boldsymbol{x} = \{x_1, \ldots, x_N\} \in \mathbb{R}^N$, the connection $\boldsymbol{x} \sim E^{\widetilde{\Psi}}$ between $\boldsymbol{x}$ and some $E^{\widetilde{\Psi}} = \widetilde{\Psi}^T E\widetilde{\Psi}$ 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 $\mathcal{P}_{\text{st}}$ for $\widetilde{A}_{\text{st}}$. Recall that the support of each $\tilde{\psi}_i(S_k(P_{j_i})$ for some $k$) is a union of patches, 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 $\widetilde{\Psi}$. Therefore we introduce the **reduced inherited energy decomposition** of $\widetilde{A}_{\text{st}} = \widetilde{\Psi}^T A\widetilde{\Psi}$ as follows:

**Definition 3.1.2** (Reduced inherited energy decomposition)**.** *With respect to the*

*underlying energy decomposition $\mathcal{E}$ of $A$, the partition $\mathcal{P}$ and the corresponding $\widetilde{\Psi}$, the* ***reduced inherited energy decomposition*** *of $\widetilde{A}_{st} = \widetilde{\Psi}^T A \widetilde{\Psi}$ is given by $\mathcal{E}_{re}^{\widetilde{\Psi}} = \{\underline{A}_{P_j}^{\widetilde{\Psi}}\}_{j=1}^{M} \cup \{E^{\widetilde{\Psi}} : E \in \mathcal{E}_{\mathcal{P}}^c\}$ with*

$$\underline{A}_{P_j}^{\widetilde{\Psi}} = \widetilde{\Psi}^T \underline{A}_{P_j} \widetilde{\Psi}, \quad j = 1, 2, \cdots, M, \text{ and} \tag{3.7}$$

$$E^{\widetilde{\Psi}} = \widetilde{\Psi}^T E \widetilde{\Psi}, \quad \forall E \in \mathcal{E}_{\mathcal{P}}^c, \tag{3.8}$$

*where $\mathcal{E}_{\mathcal{P}}^c = \mathcal{E} \backslash \mathcal{E}_{\mathcal{P}}$ with $\mathcal{E}_{\mathcal{P}} = \{E \in \mathcal{E} : \exists P_j \in \mathcal{P} \ s.t. \ E \in P_j\}$.*

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

### 3.1.3 Multiresolution Matrix 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 $\mathcal{P}^{(k)}$, $\Phi^{(k)}$, $\Psi^{(k)}$ 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

$$\begin{aligned}(A^{(k-1)})^{-1} = {} & U^{(k)}((U^{(k)})^T A^{(k-1)} U^{(k)})^{-1}(U^{(k)})^T \\ & + \Psi^{(k)}((\Psi^{(k)})^T A^{(k-1)} \Psi^{(k)})^{-1}(\Psi^{(k)})^T,\end{aligned}$$

and then define $A^{(k)} = (\Psi^{(k)})^T A^{(k-1)} \Psi^{(k)}$, $B^{(k)} = (U^{(k)})^T A^{(k-1)} U^{(k)}$, and $\mathcal{E}^{(k)} = (\mathcal{E}^{(k-1)})_{re}^{\Psi^{(k)}}$ as in Definition 3.1.2. Moreover, if we write

$$\mathbf{\Phi}^{(1)} = \Phi^{(1)}, \quad \mathbf{\Phi}^{(k)} = \Phi^{(1)}\Phi^{(2)} \cdots \Phi^{(k-1)}\Phi^{(k)}, \quad k \geq 1, \tag{3.9a}$$

$$\mathcal{U}^{(1)} = U^{(1)}, \quad \mathcal{U}^{(k)} = \Psi^{(1)}\Psi^{(2)} \cdots \Psi^{(k-1)} U^{(k)}, \quad k \geq 1, \tag{3.9b}$$

$$\mathbf{\Psi}^{(1)} = \Psi^{(1)}, \quad \mathbf{\Psi}^{(k)} = \Psi^{(1)}\Psi^{(2)} \cdots \Psi^{(k-1)}\Psi^{(k)}, \quad k \geq 1, \tag{3.9c}$$

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

$$A^{(k)} = (\mathbf{\Psi}^{(k)})^T A \mathbf{\Psi}^{(k)} = ((\mathbf{\Phi}^{(k)})^T A^{-1} \mathbf{\Phi}^{(k)})^{-1}, \quad B^{(k)} = (\mathcal{U}^{(k)})^T A \mathcal{U}^{(k)},$$

$$(\mathbf{\Phi}^{(k)})^T \mathbf{\Phi}^{(k)} = (\mathbf{\Phi}^{(k)})^T \mathbf{\Psi}^{(k)} = I_{N^{(k)}}, \quad \mathbf{\Psi}^{(k)} = A^{-1}\mathbf{\Phi}^{(k)}((\mathbf{\Phi}^{(k)})^T A^{-1} \mathbf{\Phi}^{(k)})^{-1},$$

and for any integer $K$,

$$\begin{aligned}A^{-1} &= (A^{(0)})^{-1} \\ &= \sum_{k=1}^{K} \mathcal{U}^{(k)}((\mathcal{U}^{(k)})^T A \mathcal{U}^{(k)})^{-1}(\mathcal{U}^{(k)})^T + \mathbf{\Psi}^{(K)}((\mathbf{\Psi}^{(K)})^T A \mathbf{\Psi}^{(K)})^{-1}(\mathbf{\Psi}^{(K)})^T.\end{aligned} \tag{3.10}$$

**Remark 3.1.3.**

- *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 $\Phi^{(k)}, \Psi^{(k)}, U^{(k)}$ are all constructed corresponding to this $\mathcal{P}^{(k)}$ in the same reduced space. Intuitively, if the average patch sizes (basis number in a patch) for partition $\mathcal{P}^{(k)}$ is $s^{(k)}$, then we have $N^{(k)} = \frac{q^{(k)}}{s^{(k)}} N^{(k-1)}$, 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 [8, 123]. But here the nested partitions are replaced by level-wisely 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 [88], equations* (3.9) *together with* (3.10) *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 $\mathbb{R}^n$ [88] as

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

where again we also use $\mathcal{U}^{(k)}$( or $\mathbf{\Psi}^{(k)}$) to denote the subspace spanned by the basis $\mathcal{U}^{(k)}$( or $\mathbf{\Psi}^{(k)}$). 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 $(B^{(k)})^{-1}, k = 1, \cdots, K$, 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 3.1.4.** *If on each level $\Phi^{(k)}$ is given by Construction 2.2.6 with integer $q^{(k)}$, then for $k \geq 1$ we have*

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

$$\lambda_{\max}(B^{(k)}) \leq \delta(\mathcal{P}^{(k-1)}, q^{(k-1)}) \lambda_{\max}\big((U^{(k)})^T U^{(k)}\big),$$
$$\lambda_{\min}(B^{(k)}) \geq \frac{1}{\varepsilon(\mathcal{P}^{(k)}, q^{(k)})^2} \lambda_{\min}\big((U^{(k)})^T U^{(k)}\big),$$

*and thus*

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

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

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

*Proof.* These results follow directly from Theorem 2.2.25 and Lemma 3.1.1. □

**Remark 3.1.5.** *The fact* $\lambda_{\min}(B^{(k)}) \gtrsim \frac{1}{\varepsilon(\mathcal{P}^{(k)}, q^{(k)})^2}$ *implies that the level k is a level with resolution of scale no greater than* $\varepsilon(\mathcal{P}^{(k)}, q^{(k)})$*, namely the space* $\mathcal{U}^{(k)}$ *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 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 $A$ lies in the effort to bound each $\varepsilon(\mathcal{P}^{(k)}, q^{(k)})^2 \delta(\mathcal{P}^{(k-1)}, q^{(k-1)})$ so that $B^{(k)}$ 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 $\gamma^{(k)} = \frac{\varepsilon(\mathcal{P}^{(k)}, q^{(k)})}{\varepsilon(\mathcal{P}^{(k-1)}, q^{(k-1)})}$, then we can write

$$\kappa(B^{(k)}) \leq \varepsilon(\mathcal{P}^{(k)}, q^{(k)})^2 \delta(\mathcal{P}^{(k-1)}, q^{(k-1)}) = (\gamma^{(k)})^2 \kappa(\mathcal{P}^{(k-1)}, q^{(k-1)}). \tag{3.11}$$

The partition condition number $\kappa(\mathcal{P}^{(k)}, q^{(k))})$ is a level-wise 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 $B^{(k)}$. 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 $\Psi^{(k)}$ are global. It would be pointless to bound the condition number of $B^{(k)}$ if we cannot take advantage of the locality/sparsity of $A$. 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.

### 3.2 MMD with Localization

Let $\widetilde{A}^{(0)} = A$, and we construct $\widetilde{A}^{(k)}, \widetilde{B}^{(k)}$ recursively from $\widetilde{A}^{(0)}$. More precisely, let $\widetilde{\mathcal{E}}^{(k-1)}$ be the underlying energy decomposition of $\widetilde{A}^{(k-1)}$, and $\mathcal{P}^{(k)}$, $\Phi^{(k)}$, $\Psi^{(k)}$

and $U^{(k)}$ be constructed corresponding to $\widetilde{A}^{(k-1)}$ and $\widetilde{\mathcal{E}}^{(k-1)}$ in space $\mathbb{R}^{N^{(k-1)}}$. We decompose $(\widetilde{A}^{(k-1)})^{-1}$ as

$$\begin{aligned}(\widetilde{A}^{(k-1)})^{-1} = U^{(k)}\big((U^{(k)})^T\widetilde{A}^{(k-1)}U^{(k)}\big)^{-1}(U^{(k)})^T \\ + \Psi^{(k)}\big((\Psi^{(k)})^T\widetilde{A}^{(k-1)}\Psi^{(k)}\big)^{-1}(\Psi^{(k)})^T.\end{aligned} \tag{3.12}$$

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

$$\widetilde{A}^{(k)} = (\widetilde{\Psi}^{(k)})^T\widetilde{A}^{(k-1)}\widetilde{\Psi}^{(k)}, \qquad \widetilde{B}^{(k)} = (U^{(k)})^T\widetilde{A}^{(k-1)}U^{(k)}, \tag{3.13}$$

and $\widetilde{\mathcal{E}}^{(k)} = (\widetilde{\mathcal{E}}^{(k-1)})_{re}^{\widetilde{\Psi}^{(k)}}$ as in Definition 3.1.2.

Similar to Corollary 3.1.4, we have the following estimates on the condition numbers of $A^{(k)}$ and $B^{(k)}$:

**Corollary 3.2.1.** *If on each level $\Phi^{(k)}$ is given by Construction 2.2.6 with integer $q^{(k)}$, and $\widetilde{\Psi}^{(k)}$ is a local approximator of $\Psi^{(k)}$ subject to localization error $\|\widetilde{\psi}_i^{(k)} - \psi_i^{(k)}\|_{A^{(k-1)}} \leq \frac{\epsilon}{\sqrt{N^{(k)}}}$, then for $k \geq 1$ we have*

$$\lambda_{\max}(\widetilde{A}^{(k)}) \leq \Big(1 + \frac{\epsilon}{\sqrt{\delta(\mathcal{P}^{(k)}, q^{(k)})}}\Big)^2\delta(\mathcal{P}^{(k)}, q^{(k)}), \qquad \lambda_{\min}(\widetilde{A}^{(k)}) \geq \lambda_{\min}(A),$$

$$\lambda_{\max}(\widetilde{B}^{(k)}) \leq \Big(1 + \frac{\epsilon}{\sqrt{\delta(\mathcal{P}^{k-1)}, q^{(k-1)})}}\Big)^2\delta(\mathcal{P}^{(k-1)}, q^{(k-1)})\lambda_{\max}\big((U^{(k)})^TU^{(k)}\big),$$

$$\lambda_{\min}(\widetilde{B}^{(k)}) \geq \frac{1}{\varepsilon(\mathcal{P}^{(k)}, q^{(k)})^2}\lambda_{\min}\big((U^{(k)})^TU^{(k)}\big),$$

*and thus*

$$\kappa(\widetilde{A}^{(k)}) \leq \Big(1 + \frac{\epsilon}{\sqrt{\delta(\mathcal{P}^{(k)}, q^{(k)})}}\Big)^2\delta(\mathcal{P}^{(k)}, q^{(k)})\|A^{-1}\|_2,$$

$$\kappa(\widetilde{B}^{(k)}) \leq \Big(1+\frac{\epsilon}{\sqrt{\delta(\mathcal{P}^{(k-1)}, q^{(k-1)})}}\Big)^2\varepsilon(\mathcal{P}^{(k)}, q^{(k)})^2\delta(\mathcal{P}^{(k-1)}, q^{(k-1)})\kappa\big((U^{(k)})^TU^{(k)}\big).$$

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

*Proof.* These results follow directly from the proof of Theorem 2.2.25, Corollary 2.2.26 and Lemma 3.1.1. □

One can indeed prove that $\delta(\mathcal{P}^{(k)}, q^{(k)}) \geq \frac{1}{\varepsilon(\mathcal{P}^{(k)}, q^{(k)})^2}$, and thus $\frac{\epsilon}{\sqrt{\delta(\mathcal{P}^{(k-1)}, q^{(k-1)})}} \leq \epsilon\varepsilon(\mathcal{P}^{(k)}, q^{(k)})$ is a small number. Therefore Corollary 3.2.1 states that the multiresolution decomposition with localization has estimates on condition numbers of

the same order as in Corollary 3.1.4, i.e. $\kappa(\widetilde{A}^{(k)}) \leq O(\delta(\mathcal{P}^{(k)}, q^{(k)})\|A^{-1}\|_2)$ and $\kappa(\widetilde{B}^{(k)}) \leq O(\varepsilon(\mathcal{P}^{(k)}, q^{(k)})^2\delta(\mathcal{P}^{(k-1)}, q^{(k-1)}))$. Having this in hand, we proceed to discuss the desired sparsity of $\widetilde{A}^{(k)}$ and $\widetilde{B}^{(k)}$.

**Locality Preservation:** Similar to the locality discussion of $\widetilde{A}_{st}$ in Section 2.3, under the locality condition (2.63), we have the following recursive estimate on the number of nonzero entries of each $A^{(k)}$ as

$$nnz(\widetilde{A}^{(k)}) = O(nnz(\widetilde{A}^{(k-1)}) \cdot (q^{(k)})^2 \cdot \frac{1}{s^{(k)}} \cdot (r^{(k)})^d), \tag{3.14}$$

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

$$nnz(\widetilde{B}^{(k)}) = O(nnz(\widetilde{A}^{(k-1)}) \cdot s^{(k)}). \tag{3.15}$$

In fact, the basis $U^{(k)}$ can be computed from $\Phi^{(k)}$ using the implicit QR factorization [102], 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 $\widetilde{B}^{(k)} = (U^{(k)})^T \widetilde{A}^{(k-1)} U^{(k)}$ (in iterative method), only the NNZ of $\widetilde{A}^{(k-1)}$ 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 $\frac{(q^{(k)})^2 (r^{(k)})^d}{s^{(k)}}$, if not well controlled, will compromise the sparsity inherited from $\widetilde{A}^{(0)} = A$. 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 $nnz(\widetilde{A}^{(k)}) = O(nnz(A))$. In particular, when we impose the localization error $\|\tilde{\psi}_i^{(k)} - \psi_i^{(k)}\|_{A^{(k-1)}} \leq \frac{\epsilon}{\sqrt{N^{(k)}}}$ on each level $k$ for some uniform $\epsilon$, we have $r^{(k)} = O(\log \frac{1}{\epsilon} + \log N^{(k)} + \log \delta(\mathcal{P}^{(k)}, q^{(k)}))$ according the discussions in Section 2.2.2. Then the sparsity condition becomes

$$o(s^{(k)}) \geq (q^{(k)})^2 \big( \log \frac{1}{\epsilon} + \log N^{(k)} + \log \delta(\mathcal{P}^{(k)}, q^{(k)}) \big)^d, \quad k \geq 1. \tag{3.16}$$

This lower bound of the patch size $s^{(k)}$ means that we need to compress enough dimensions from higher level to lower level in order to preserve sparsity due to the outreaching support of the localized basis $\widetilde{\Psi}^{(k)}$.

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 $\delta(\mathcal{P}^{(k)}, q^{(k)}) \leq \frac{c}{\varepsilon(\mathcal{P}^{(k)}, q^{(k)})^2} \leq \frac{c}{\epsilon^2}$. Therefore a safe uniform criterion for patch size $s^{(k)}$ is

$$O(s^{(k)}) = s = q^2(\log \frac{1}{\epsilon} + \log n)^{d+l}, \tag{3.17}$$

for some small $l > 0$, which asymptotically, when $n$ goes large and $\epsilon$ goes small, will ensure $nnz(A^{(k)}) = O(nnz(A))$ down through the decomposition. Since the decomposition should stop when $N^{(K)}$, the dimension of $A^{(K)}$ is small enough, namely when $n = O((s/q)^K)$, (3.17) also gives us an estimate of the total level number as

$$K = O(\log_{s/q} n) = O(\frac{\log n}{\log(q(\log \frac{1}{\epsilon} + \log n)^{d+l})}) = O(\frac{\log n}{\log(\log \frac{1}{\epsilon} + \log n)}). \tag{3.18}$$

**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 (2.63), and an intuitive geometry estimate gives $\frac{S^{(k)}}{S^{(k-1)}} = s^{(k)}$. As a consequence, under the local energy decomposition condition (2.64) of order $(q, p)$, we have the following estimate

$$\gamma^{(k)} = \frac{\varepsilon(\mathcal{P}^{(k)}, q)}{\varepsilon(\mathcal{P}^{(k-1)}, q)} = O(\Big(\frac{R^{(k)}}{R^{(k-1)}}\Big)^p) = O(\Big(\frac{S^{(k)}}{S^{(k-1)}}\Big)^{\frac{p}{d}}) = O((s^{(k)})^{\frac{p}{d}}). \tag{3.19}$$

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 [53]. Under the sparsity condition (3.16) and considering $q$ as a constant, we require

$$o(\gamma^{(k)}) \geq (\log \frac{1}{\epsilon} + \log N^{(k)} + \log \delta(\mathcal{P}^{(k)}, q^{(k)}))^p, \tag{3.20}$$

to ensure the sparsity of the decomposition, and similarly a safe, uniform choice of the scale ratio $\gamma^{(k)}$ is

$$\gamma^{(k)} = \gamma = (\log \frac{1}{\epsilon} + \log n)^{p+l}, \tag{3.21}$$

for some small $l > 0$. Such choice provides a uniform bound on the condition number of $\widetilde{B}^{(k)}$ as

$$\kappa(\widetilde{B}^{(k)}) \leq O(\varepsilon(\mathcal{P}^{(k)}, q^{(k)})^2 \delta(\mathcal{P}^{(k-1)}, q^{(k-1)})) \leq O((\log \frac{1}{\epsilon} + \log n)^{p+l}) \tag{3.22}$$

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 (3.22) is valid for $k \geq 2$. For consistency, we choose $\varepsilon(\mathcal{P}^{(1)}, q^{(1)})^2 = O(\frac{(\log \frac{1}{\epsilon} + \log n)^{p+l}}{\|A\|_2})$ so that (3.22) is also valid for $k = 1$.

**Remark 3.2.2.** *By estimate* (3.22)*, the bound on* $\kappa(\widetilde{B}^{(k)})$ *will go to infinity when n goes to infinity. In our construction of the multiresolution decomposition for resolving large condition number of A, we cannot asymptotically have an absolute constant bound for* $\kappa(\widetilde{B}^{(k)})$ *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* $\widetilde{A}^{(k)}$ *is based on the sparsity of* $\widetilde{A}^{(k-1)}$*, as shown in* (3.14)*. However, in [89], the existence of nested measurement function* $\Phi$ *is assumed a-priori before the construction of the multiresolution structure, and thus the sparsity of* $\widetilde{A}^{(k)}$ *can be inherited directly from* $\widetilde{A}^{(0)}$*, which avoids the accumulation of the factor* $\frac{(q^{(k)})^2 (r^{(k)})^d}{s^{(k)}}$ *through levels. As a result, the sparsity of* $\widetilde{A}^{(k)}$ *does not contradict the uniform bound of* $\kappa(\widetilde{B}^{(k)})$*.*

### 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 $\Psi^{(k)}((\Psi^{(k)})^T \widetilde{A}^{(k-1)} \Psi^{(k)})^{-1} (\Psi^{(k)})^T$ and $\widetilde{\Psi}^{(k)}((\widetilde{\Psi}^{(k)})^T \widetilde{A}^{(k-1)} \widetilde{\Psi}^{(k)})^{-1} (\widetilde{\Psi}^{(k)})^T$; (ii) the error caused by solving $(A^{(k)})^{-1} = ((\widetilde{\Psi}^{(k)})^T \widetilde{A}^{(k-1)} \widetilde{\Psi}^{(k)})^{-1}$ (or $(\widetilde{B}^{(k)})^{-1}$) 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 3.2.3.** *Given an integer K, let* $\mathrm{Inv}(A)$ *denote the solver for* $A^{-1}$ *using K-level's multiresolution operator decomposition with localization. Assume that*

*(i) each* $(\widetilde{B}^{(k)})^{-1}$ *can be solved efficiently subject to a uniform relative error bound* $\mathrm{err}_B$ *in the sense that the solver* $\mathrm{Inv}(\widetilde{B}^{(k)})$ *(as a linear operator) satisfies*

$$\|(\widetilde{B}^{(k)})^{-1} b - \mathrm{Inv}(\widetilde{B}^{(k)}) b\|_{\widetilde{B}^{(k)}} \leq \mathrm{err}_B \|b\|_{(\widetilde{B}^{(k)})^{-1}}, \quad \forall b \in \mathbb{R}^{N^{(k)}}, \quad \forall 1 \leq k \leq K; \tag{3.23}$$

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

$$\|(\widetilde{A}^{(K)})^{-1} b - \mathrm{Inv}(\widetilde{A}^{(K)}) b\|_{\widetilde{A}^{(K)}} \leq \mathrm{err}_A^{(K)} \|b\|_{(\widetilde{A}^{(K)})^{-1}}, \quad \forall b \in \mathbb{R}^{N^{(K)}}. \tag{3.24}$$

*(iii) each* $\widetilde{\Psi}^{(k)}$ *satisfies the localization approximation property*

$$\|\widetilde{\psi}_i^{(k)} - \psi_i^{(k)}\|_{\widetilde{A}^{(k-1)}} \leq \frac{\mathrm{err}_{loc}}{2\sqrt{N^{(k)}\|A^{-1}\|_2}}, \quad 1 \leq i \leq N^{(k)}, \tag{3.25}$$

*with a uniform constant* $\mathrm{err}_{loc}$.

*Then we have*

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

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

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

*where*

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

*Proof.* First by assumption (ii), we have

$$\|(\widetilde{A}^{(K)})^{-1}b - \mathrm{Inv}(\widetilde{A}^{(k)})b\|_{\widetilde{A}^{(K)}} \leq \mathrm{err}_A^{(K)}\|b\|_{(\widetilde{A}^{(K)})^{-1}}, \quad \forall b \in \mathbb{R}^{N^{(K)}}.$$

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

$$\|(\widetilde{A}^{(k)})^{-1}b - \mathrm{Inv}(\widetilde{A}^{(k)})b\|_{\widetilde{A}^{(k)}} \leq \mathrm{err}_A^{(k)}\|b\|_{(\widetilde{A}^{(k)})^{-1}}, \quad \forall b \in \mathbb{R}^{N^{(k)}}.$$

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

$$(\widetilde{A}^{(k-1)})^{-1} = U^{(k)}\big((U^{(k)})^T\widetilde{A}^{(k-1)}U^{(k)}\big)^{-1}(U^{(k)})^T + \Psi^{(k)}\big((\Psi^{(k)})^T\widetilde{A}^{(k-1)}\Psi^{(k)}\big)^{-1}(\Psi^{(k)})^T, \tag{3.26}$$

$$\mathrm{Inv}(\widetilde{A}^{(k-1)}) = U^{(k)}\mathrm{Inv}(\widetilde{B}^{(k)})(U^{(k)})^T + \widetilde{\Psi}^{(k)}\mathrm{Inv}(\widetilde{A}^{(k)})(\widetilde{\Psi}^{(k)})^T, \tag{3.27}$$

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

$$\begin{aligned}
&\|(\widetilde{A}^{(k-1)})^{-1}b - \mathrm{Inv}(\widetilde{A}^{(k)})b\|_{\widetilde{A}^{(k-1)}} \\
\leq\ &\|U^{(k)}\big((U^{(k)})^T\widetilde{A}^{(k-1)}U^{(k)}\big)^{-1}(U^{(k)})^Tb - U^{(k)}\mathrm{Inv}(\widetilde{B}^{(k)})(U^{(k)})^Tb\|_{\widetilde{A}^{(k-1)}} \\
&+ \|\Psi^{(k)}\big((\Psi^{(k)})^T\widetilde{A}^{(k-1)}\Psi^{(k)}\big)^{-1}(\Psi^{(k)})^Tb - \widetilde{\Psi}^{(k)}\big((\widetilde{\Psi}^{(k)})^T\widetilde{A}^{(k-1)}\widetilde{\Psi}^{(k)}\big)^{-1}(\widetilde{\Psi}^{(k)})^Tb\|_{\widetilde{A}^{(k-1)}} \\
&+ \|\widetilde{\Psi}^{(k)}\big((\widetilde{\Psi}^{(k)})^T\widetilde{A}^{(k-1)}\widetilde{\Psi}^{(k)}\big)^{-1}(\widetilde{\Psi}^{(k)})^Tb - \widetilde{\Psi}^{(k)}\mathrm{Inv}(\widetilde{A}^{(k)})(\widetilde{\Psi}^{(k)})^Tb\|_{\widetilde{A}^{(k-1)}} \\
=\ &I_1 + I_2 + I_3.
\end{aligned}$$

Recall that $\widetilde{A}^{(k)} = (\widetilde{\Psi}^{(k)})^T \widetilde{A}^{(k-1)} \widetilde{\Psi}^{(k)}$, $\widetilde{B}^{(k)} = (U^{(k)})^T \widetilde{A}^{(k-1)} U^{(k)}$, then by assumption (i) we have

$$\begin{aligned} I_1 &= \|(\widetilde{B}^{(k)})^{-1}(U^{(k)})^T b - \mathrm{Inv}(\widetilde{B}^{(k)})(U^{(k)})^T b\|_{\widetilde{B}^{(k)}} \\ &\leq \mathrm{err}_B \Big( b^T U^{(k)} (\widetilde{B}^{(k)})^{-1} (U^{(k)})^T b \Big)^{\frac{1}{2}} \\ &\leq \mathrm{err}_B \|(\widetilde{A}^{(k-1)})^{\frac{1}{2}} U^{(k)} ((U^{(k)})^T \widetilde{A}^{(k-1)} U^{(k)})^{-1} (U^{(k)})^T (\widetilde{A}^{(k-1)})^{\frac{1}{2}} \|_2 \|b\|_{(\widetilde{A}^{(k-1)})^{-1}} \\ &\leq \mathrm{err}_B \|b\|_{(\widetilde{A}^{(k-1)})^{-1}}. \end{aligned}$$

Similarly by the assumption of induction, we have

$$I_3 = \|(\widetilde{A}^{(k)})^{-1}(\widetilde{\Psi}^{(k)})^T b - \mathrm{Inv}(\widetilde{A}^{(k)})(\widetilde{\Psi}^{(k)})^T b\|_{\widetilde{A}^{(k)}} \leq \mathrm{err}_A^{(k)} \|b\|_{(\widetilde{A}^{(k-1)})^{-1}}.$$

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

$$\begin{aligned} I_2 &= \|P_{\Psi^{(k)}}^{\widetilde{A}^{(k-1)}} x - P_{\widetilde{\Psi}^{(k)}}^{\widetilde{A}^{(k-1)}} x\|_{\widetilde{A}^{(k-1)}} \\ &\leq \|P_{\Psi^{(k)}}^{\widetilde{A}^{(k-1)}} x - P_{\widetilde{\Psi}^{(k)}}^{\widetilde{A}^{(k-1)}} P_{\Psi^{(k)}}^{\widetilde{A}^{(k-1)}} x\|_{\widetilde{A}^{(k-1)}} + \|P_{\widetilde{\Psi}^{(k)}}^{\widetilde{A}^{(k-1)}} P_{U^{(k)}}^{\widetilde{A}^{(k-1)}} x\|_{\widetilde{A}^{(k-1)}}. \end{aligned}$$

Using a similar argument in Lemma 2.2.8 we can actually prove by assumption (iii) that

$$\|P_{\Psi^{(k)}}^{\widetilde{A}^{(k-1)}} x - P_{\widetilde{\Psi}^{(k)}}^{\widetilde{A}^{(k-1)}} P_{\Psi^{(k)}}^{\widetilde{A}^{(k-1)}} x\|_{\widetilde{A}^{(k-1)}} \leq \frac{1}{2} \mathrm{err}_{loc} \|P_{\Psi^{(k)}}^{\widetilde{A}^{(k-1)}} x\|_{\widetilde{A}^{(k-1)}} \leq \frac{1}{2} \mathrm{err}_{loc} \|b\|_{(\widetilde{A}^{(k-1)})^{-1}},$$

and

$$\begin{aligned} &\|P_{\widetilde{\Psi}^{(k)}}^{\widetilde{A}^{(k-1)}} P_{U^{(k)}}^{\widetilde{A}^{(k-1)}} x\|_{\widetilde{A}^{(k-1)}} \\ &\leq \|((\widetilde{\Psi}^{(k)})^T \widetilde{A}^{(k-1)} \widetilde{\Psi}^{(k)})^{-1}\|_2^{\frac{1}{2}} \|(\widetilde{\Psi}^{(k)})^T \widetilde{A}^{(k-1)} P_{U^{(k)}}^{\widetilde{A}^{(k-1)}} x\|_2 \\ &\leq \|A^{-1}\|_2^{\frac{1}{2}} \|(\widetilde{\Psi}^{(k)} - \Psi^{(k)})^T \widetilde{A}^{(k-1)} P_{U^{(k)}}^{\widetilde{A}^{(k-1)}} x\|_2 \\ &\leq \|A^{-1}\|_2^{\frac{1}{2}} \|(\widetilde{\Psi}^{(k)} - \Psi^{(k)})^T \widetilde{A}^{(k-1)} (\widetilde{\Psi}^{(k)} - \Psi^{(k)})\|_2^{\frac{1}{2}} \|P_{U^{(k)}}^{\widetilde{A}^{(k-1)}} x\|_{\widetilde{A}^{(k-1)}} \\ &\leq \frac{1}{2} \mathrm{err}_{loc} \|b\|_{(\widetilde{A}^{(k-1)})^{-1}}, \end{aligned}$$

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

$$\|(\widetilde{A}^{(k-1)})^{-1} b - \mathrm{Inv}(\widetilde{A}^{(k-1)}) b\|_{\widetilde{A}^{(k-1)}} \leq (\mathrm{err}_B + \mathrm{err}_{loc} + \mathrm{err}_A^{(k)}) \|b\|_{(\widetilde{A}^{(k-1)})^{-1}},$$

that is we have

$$\mathrm{err}_A^{(k-1)} = (\mathrm{err}_B + \mathrm{err}_{loc} + \mathrm{err}_A^{(k)}).$$

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

$$\mathrm{err}_{total} = \mathrm{err}_A^{(0)} = K(\mathrm{err}_B + \mathrm{err}_{loc}) + \mathrm{err}_A^{(K)}.$$

□

**Remark 3.2.4.** *Assumption (i) is reasonable since each $\widetilde{B}^{(k)}$ inherit the sparsity from A, and its condition numbers can be well bounded in order $O((\log \frac{1}{\mathrm{err}_{loc}} + \log n)^{p+l})$. Assumption* ((ii)) *is reasonable since each $\widetilde{A}^{(K)}$ is of small dimension when K is as large as in* (3.18)*. Assumption (iii) is reasonable due to the exponential decay property of each $\Psi^{(k)}$. 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 Section 3.4.*

$$\mathcal{E}^{(k-1)} \xrightarrow{\text{Algorithm 4}} \mathcal{P}(k) \xrightarrow{\text{Algorithm 2}} \Phi(k) \xrightarrow{\text{Algorithm 3}} \widetilde{\Psi}(k) \xrightarrow{\substack{\text{Inherited Energy}\\ \text{decomposition}}} \mathcal{E}^{(k)}$$

Figure 3.1: Process flowchart of Algorithm 6.

### 3.2.1 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 3.1 shows the flowchart of Algorithm 6.

**Algorithm 6** *MMD with localization*

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

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

1: **for** $k = 1 : K$ **do**
2: Construct $\mathcal{P}^{(k)}, \Phi^{(k)}, U^{(k)}, \widetilde{\Psi}^{(k)}$ with Algorithm 1, with respect to $\widetilde{A}^{(k-1)}, \widetilde{\mathcal{E}}^{(k-1)}$, and subject to $q^{(k)}, \varepsilon^{(k)}, c^{(k)}$ and localization error $\frac{\epsilon}{\sqrt{N^{(k)}}}$;
3: Compute $\widetilde{A}^{(k)}$ and $\widetilde{B}^{(k)}$ by (3.13);
4: Compute reduced energy $\widetilde{\mathcal{E}}^{(k)}$ by (3.8);
5: output/store $\widetilde{\Psi}^{(k)}, U^{(k)}, \widetilde{B}^{(k)}$;
6: **end for**
7: output/store $\widetilde{A}^{(K)}$.

**Remark 3.2.5.**

- *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 posterior omit the level-wise energy decompositions and partitions. Then if we see our*

**Algorithm 7** *Solving linear system with MMD with localization*

**Input:** $\widetilde{\Psi}^{(k)}, U^{(k)}, \widetilde{B}^{(k)}$ for $k = 1, 2, \cdots, K$, $\widetilde{A}^{(K)}$, load vector $b = b^{(0)}$, prescribed relative accuracy $\epsilon$

**Output:** Approximated solution $x^{(0)}$.

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

*level-wisely constructed* $\Phi$ *as a nested sequence* (3.9a)*, our decomposition is structurally equivalent to the result obtained in [89], where the existence of such nested* $\Phi$ *is a priori assumed. Therefore, the required properties of the nested sequence in Condition 2.3 of [89] are similar to the assumption in Theorem 3.2.3.*

**Complexity of Algorithm 6**

Assume that locality conditions (2.63),(2.64),(2.65) are true with constant $d, p, q, c$. Then all $q^{(k)}$ and $c^{(k)}$ are chosen uniformly over levels to be $q, c$ respectively. $\varepsilon^{(k)}$ is chosen subject to scale ratio choice (3.21), $(\varepsilon^{(1)})^2 = \frac{(\log \frac{1}{\epsilon} + \log n)^{p+l}}{\|A\|_2}$ 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 (3.22). Then the complexity of Line 2 can be modified from (2.69) as

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

where $s = O((\log \frac{1}{\epsilon} + \log n)^{d(1+l/p)})$ according to estimate (3.19). The complexity of Line 3 and 4(sparse matrices multiplication) together can be bounded by

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

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

$$O(d \cdot s^2 \cdot \log s \cdot n) + O(\log s \cdot n \cdot \log n) + O(q \cdot n \cdot (\log \frac{1}{\epsilon} + \log n)^{3d+p}), \quad (3.28)$$

where we have assumed that $d \geq 1 \geq l$. The total complexity of Algorithm 6 is then the level number $K$ times (3.28). By (3.18), we have $K = O(\frac{\log n}{\log(\log \frac{1}{\epsilon} + \log n)}) \leq O(\log n)$ and $K \log s = O(\log n)$. Thus the total complexity of Algorithm 6 is

$$\begin{aligned} &O(d \cdot s^2 \cdot \log n \cdot n) + O(n \cdot (\log n)^2) + O(K \cdot q \cdot n \cdot (\log \frac{1}{\epsilon} + \log n)^{3d+p}) \\ &\leq O(m \cdot \log n \cdot (\log \frac{1}{\epsilon} + \log n)^{3d+p}). \end{aligned} \tag{3.29}$$

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 $\widetilde{A}^{(k)}$ is bounded by $O(nnz(A)) = O(m)$, and the condition number each $\widetilde{B}^{(k)}$ can be bounded by $O((\log \frac{1}{\epsilon} + \log n)^{p+l})$, then the complexity of solving linear system in Line 3 using a CG type method is bounded by

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

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 level-wise relative accuracy $\epsilon$ is

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

However, by Theorem 3.2.3, the total accuracy is $\epsilon_{total} = O(K\epsilon)$. Thus the complexity of Algorithm 7 subject to a total relative accuracy $\epsilon_{total}$ is

$$O(m \cdot (\log \frac{1}{\epsilon_{total}} + \log n)^{p+l} \cdot (\log \frac{1}{\epsilon_{total}} + \log \log n) \cdot \log n). \tag{3.30}$$

## 3.3 Multilevel Operator Compression

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

$$A^{-1} \approx \mathbf{\Psi}^{(K)} \big( (\mathbf{\Psi}^{(K)})^T A \mathbf{\Psi}^{(K)} \big)^{-1} (\mathbf{\Psi}^{(K)})^T = P^A_{\mathbf{\Psi}^{(K)}} A^{-1}. \tag{3.31}$$

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 $\mathbf{\Phi}^{(k)}$ 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 (3.31) is accumulated over all levels finer than level $K$. To quantify such compression error, we introduce the following theorem:

**Theorem 3.3.1.** *Assume that on each level $\Phi^{(k)}$ is given by Construction 2.2.6 with integer $q^{(k)}$. Then we have*

$$\|x - P_{\mathbf{\Phi}^{(K)}} x\|_2 \leq \Big( \sum_{k=0}^{K} \varepsilon(\mathcal{P}^{(k)}, q^{(k)})^2 \Big)^{\frac{1}{2}} \|x\|_A, \quad \forall x \in \mathbb{R}^n, \tag{3.32}$$

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

$$\|x - P^A_{\mathbf{\Psi}^{(K)}} x\|_A \leq \Big( \sum_{k=1}^{K} \varepsilon(\mathcal{P}^{(k)}, q^{(k)})^2 \Big)^{\frac{1}{2}} \|b\|_2,$$

$$\|x - P^A_{\mathbf{\Psi}^{(K)}} x\|_2 \leq \Big( \sum_{k=1}^{K} \varepsilon(\mathcal{P}^{(k)}, q^{(k)})^2 \Big) \|b\|_2,$$

$$\|A^{-1} - P^A_{\mathbf{\Psi}^{(K)}} A^{-1}\|_2 \leq \Big( \sum_{k=1}^{K} \varepsilon(\mathcal{P}^{(k)}, q^{(k)})^2 \Big).$$

*Proof.* Again by Theorem 2.2.1, we only need to prove (3.32). For consistency, we write $\mathbf{\Phi}^{(0)} = I_n$, and correspondingly $\mathbf{\Psi}^{(0)} = I_n$, $P_{\mathbf{\Phi}^{(0)}} = I_n$, $P^A_{\mathbf{\Psi}^{(0)}} = I_n$. Using (3.9), it is easy to check that for any $x \in \mathbb{R}^n$ and any $k_1 \leq k_2 \leq k_3$,

$$(P_{\mathbf{\Phi}^{(k_1)}} x - P_{\mathbf{\Phi}^{(k_2)}} x)^T (P_{\mathbf{\Phi}^{(k_2)}} x - P_{\mathbf{\Phi}^{(k_3)}} x) = 0,$$

thus we have

$$\|x - P_{\mathbf{\Phi}^{(K)}} x\|_2^2 = \| \sum_{k=1}^{K} (P_{\mathbf{\Phi}^{(k-1)}} x - P_{\mathbf{\Phi}^{(k)}} x)\|_2^2 = \sum_{k=1}^{K} \|P_{\mathbf{\Phi}^{(k-1)}} x - P_{\mathbf{\Phi}^{(k)}} x\|_2^2.$$

Notice that

$$P_{\mathbf{\Phi}^{(k-1)}} x - P_{\mathbf{\Phi}^{(k)}} x = \mathbf{\Phi}^{(k-1)} (\mathbf{\Phi}^{(k-1)})^T x - \mathbf{\Phi}^{(k-1)} \Phi^{(k)} (\Phi^{(k)})^T (\mathbf{\Phi}^{(k-1)})^T x,$$

thus by the construction of $\Phi^{(k)}$( or $\mathbf{\Phi}^{(k)}$), we have

$$\begin{aligned}
\|P_{\mathbf{\Phi}^{(k-1)}} x - P_{\mathbf{\Phi}^{(k)}} x\|_2^2 &= \|(\mathbf{\Phi}^{(k-1)})^T x - \Phi^{(k)} (\Phi^{(k)})^T (\mathbf{\Phi}^{(k-1)})^T x\|_2^2 \\
&\leq \varepsilon(\mathcal{P}^{(k)}, q^{(k)})^2 \|(\mathbf{\Phi}^{(k-1)})^T x\|^2_{A^{(k-1)}} \\
&= \varepsilon(\mathcal{P}^{(k)}, q^{(k)})^2 x^T \mathbf{\Phi}^{(k-1)} \big( (\mathbf{\Phi}^{(k-1)})^T A^{-1} \mathbf{\Phi}^{(k-1)} \big)^{-1} (\mathbf{\Phi}^{(k-1)})^T x \\
&= \varepsilon(\mathcal{P}^{(k)}, q^{(k)})^2 \|P^A_{\mathbf{\Psi}^{(k-1)}} x\|_A^2 \\
&\leq \varepsilon(\mathcal{P}^{(k)}, q^{(k)})^2 \|x\|_A^2.
\end{aligned}$$

We have used the fact that

$$\begin{aligned}\|P^A_{\mathbf{\Psi}^{(k)}}x\|_A^2 &= x^T A\mathbf{\Psi}^{(k)}((\mathbf{\Psi}^{(k)})^T A\mathbf{\Psi}^{(k)})^{-1}(\mathbf{\Psi}^{(k)})^T Ax \\ &= x^T\mathbf{\Phi}^{(k)}((\mathbf{\Phi}^{(k)})^T A^{-1}\mathbf{\Phi}^{(k)})^{-1}(\mathbf{\Phi}^{(k)})^T x, \quad \forall k \geq 0.\end{aligned}$$

Therefore we have

$$\|x - P_{\mathbf{\Phi}^{(K)}}x\|_2^2 \leq \Big(\sum_{k=0}^{K}\varepsilon(\mathcal{P}^{(k)}, q^{(k)})^2\Big)\|x\|_A^2.$$

□

**Remark 3.3.2.**

- *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*

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

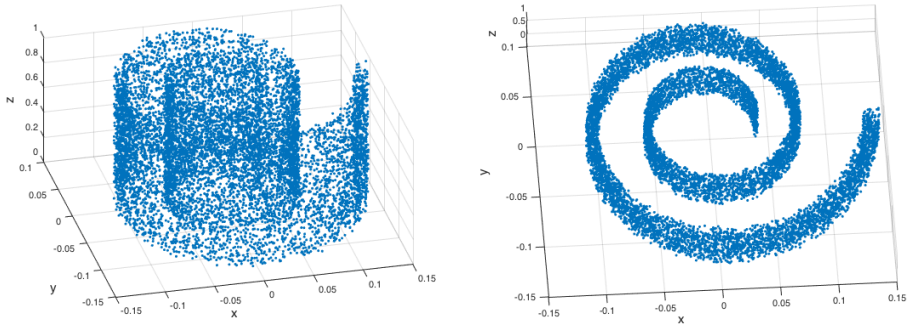
  *which is an error of scale* $\varepsilon(\mathcal{P}^{(K)}, q^{(K)})^2$ *as we expected.*

- *Again one shall be aware of the difference between the one-level compression with* ***error factor*** $\varepsilon(\mathcal{P}^{(K)}, q^{(K)})$ *and the multi-level compression in Section 3.1.3. A one-level compression with* ***error factor*** $\varepsilon(\mathcal{P}^{(K)}, q^{(K)})$ *requires to construct* $\mathcal{P}^{(K)}, \mathbf{\Phi}^{(k)}$ *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* $\mathcal{P}^{(K)}$ *when* $\varepsilon(\mathcal{P}^{(K)}, q^{(K)})$ *is a coarse scale. But the multi-level compression in Section 3.1.3 is computed hierarchically with bounded compression ratio between levels, and thus only involves eigenvalue problems on patches of well-bounded size(*$s = O(\log\frac{1}{\epsilon} + \log n)^{d+l}$*) 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 3.3.1.*

### 3.4 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 the Example 1 in Section 2.4.1, with density factor $\eta = 2$. But this time the vertices of the graph



(a) Front view (b) Top view

Figure 3.2: A "Roll surface" constructed by (3.33).

are randomly distributed around a 2-dimensional roll surface of area 1 in $\mathbb{R}^3$. The distribution is a combination of a uniform distribution over the surface and an up to 10% random displacement off the surface. More precisely, the two-dimensional roll is characterized as

$$(x(t), y(t), z) = (\rho(t)\cos(\theta(t)), \rho(t)\sin(\theta(t)), z), \quad t \in [0,1], z \in [0,1],$$

where

$$\theta(t) = \frac{1}{a}\log\left(1 + t(e^{4\pi a} - 1)\right), \quad \rho(t) = \frac{a}{\sqrt{1+a^2}}(t + \frac{1}{e^{4\pi a} - 1}),$$

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

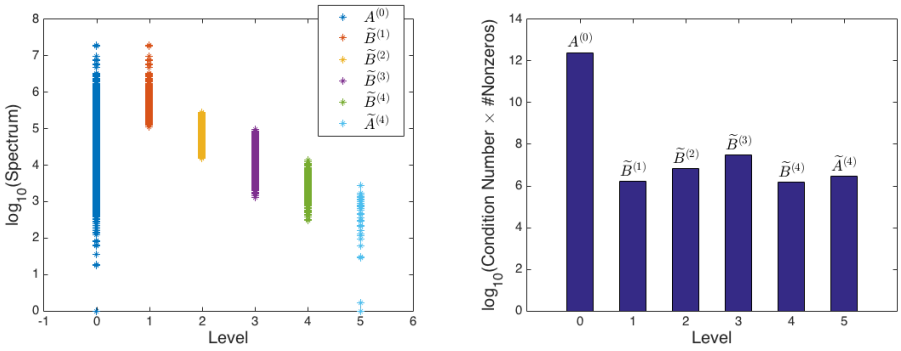
$$(x_i, y_i, z_i) = (\eta_i\rho(t_i)\cos(\theta(t_i)), \eta_i\rho(t_i)\sin(\theta(t_i)), z_i), \quad i = 1, 2, \cdots, n, \tag{3.33}$$

where $t_i \overset{i.i.d}{\sim} \mathcal{U}[0,1]$, $z_i \overset{i.i.d}{\sim} \mathcal{U}[0,1]$, and $\eta_i \overset{i.i.d}{\sim} \mathcal{U}[0.9, 1.1]$. In this example we take $n = 10000$ and $a = 0.1$. Figure 3.2a and Figure 3.2b show the point cloud of all vertices. This explicit expression, however, is considered as a hidden geometric information, and is not employed in our partitioning algorithm.

The Laplacian $L = A_0$ and the energy decomposition $\mathcal{E}$ are given as in Example 2.1.10, and we apply a 4-level multiresolution matrix decomposition with localization using Algorithm 6 to decompose the problem of solving $L^{-1}$. In this particular case we have $\lambda_{\max}(L) = 1.93 \times 10^7$ and $\lambda_{\min}(L) = 1$. Again for graph Laplacian, we choose $q = 1$. On each level $k$, the partition is constructed subject to $\varepsilon(\mathcal{P}^{(k)}, 1)^2 = 10^{k-6}$ (i.e. $\{\varepsilon(\mathcal{P}^{(k)}, 1)^2\}_{k=1}^4 = \{0.00001, 0.0001, 0.001, 0.01\}$) and $\delta(\mathcal{P}^{(k)}, 1)\varepsilon(\mathcal{P}^{(k)}, 1)^2 \leq 50$. The compliment space $U^{(k)}$ are extended from

| Level | Size | #Nonzeros | Condition Number | Complexity |
|---|---|---|---|---|
| $L$ | $10000 \times 10000$ | $128018 \triangleq m$ | $1.93 \times 10^7$ | $2.47 \times 10^{12}$ |
| $\widetilde{B}^{(1)}$ | $1898 \times 1898$ | $9812 \approx 0.07m$ | $1.72 \times 10^2$ | $1.69 \times 10^6$ |
| $\widetilde{B}^{(2)}$ | $6639 \times 6639$ | $391499 \approx 3.06m$ | $1.80 \times 10^1$ | $7.04 \times 10^6$ |
| $\widetilde{B}^{(3)}$ | $1244 \times 1244$ | $417156 \approx 3.26m$ | $7.27 \times 10^1$ | $3.03 \times 10^7$ |
| $\widetilde{B}^{(4)}$ | $186 \times 186$ | $34596 \approx 0.27m$ | $4.47 \times 10^1$ | $1.55 \times 10^6$ |
| $\widetilde{A}^{(4)}$ | $33 \times 33$ | $1025 \approx 0.008m$ | $2.83 \times 10^3$ | $2.90 \times 10^6$ |
| total | - | $854088 \approx 6.67m$ | - | $4.34 \times 10^7$ |

Table 3.1: Complexity results of 4-level MMD with localization using Algorithm 6.




(a) Spectrum (b) Complexity

Figure 3.3: Spectrum and complexity of each layer in a 4-level MMD obtained from Algorithm 6.

$\Phi^{(k)}$ using a patch-wise QR factorization. So according to Corollary 3.2.1, each $\kappa(\widetilde{B}^{(k)})$, $k = 2, 3, 4$ is expected to be bounded by

$$\delta(\mathcal{P}^{(k-1)}, 1)\varepsilon(\mathcal{P}^{(k)}, 1)^2 = \delta(\mathcal{P}^{(k-1)}, 1)\varepsilon(\mathcal{P}^{(k-1)}, 1)^2 \frac{\varepsilon(\mathcal{P}^{(k)}, 1)^2}{\varepsilon(\mathcal{P}^{(k-1)}, 1)^2} \leq 500,$$

$\kappa(\widetilde{B}^{(1)})$ is expected bounded by $\lambda_{\max}(L)\varepsilon(\mathcal{P}^{(0)}, 1)^2 = 1.93 \times 10^2$, and $\kappa(\widetilde{A}^{(4)})$ is expected to be bounded by

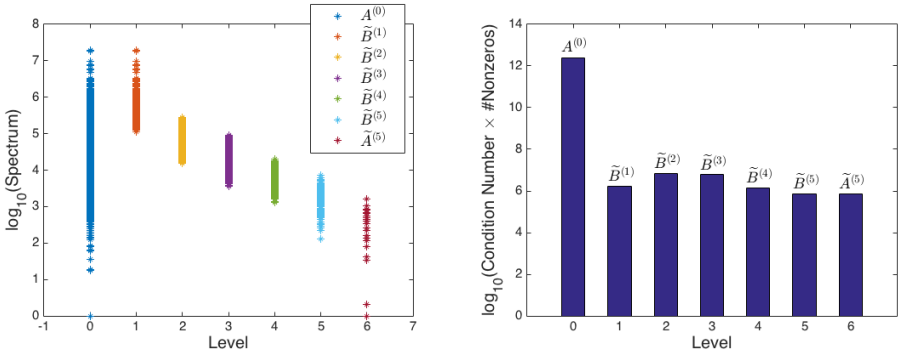
$$\delta(\mathcal{P}^{(3)}, 1)\lambda_{\min}(L)^{-1} = \delta(\mathcal{P}^{(3)}, 1)\varepsilon(\mathcal{P}^{(3)}, 1)^2 \frac{\lambda_{\min}(L)^{-1}}{\varepsilon(\mathcal{P}^{(3)}, 1)^2} \leq 5000.$$

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 number of nonzero (NNZ) entries and the condition number of the matrix concerned, given a fixed prescribed relative accuracy [102]. 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 [102]. The results not only satisfy the theoretical prediction, but also turn out to be much better than expected as shown in Table 3.1 and Figure 3.3. We

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

Table 3.2: Complexity results of a 5-level MMD with localization using Algorithm 6.




(a) (b)

Figure 3.4: Spectrum and complexity of each layer in a 5-level MMD obtained from Algorithm 6.

now verify the performance of the 4-level and the 5-level MMDs by solving two particular systems $Lu^* = b$, where

$$\text{Case 1: } u_i^* = (x_i^2 + y_i^2 + z_i^2)^{\frac{1}{2}},\ i = 1, 2, \cdots, n, b = Lu^*;$$
$$\text{Case 2: } u_i^* = x_i + y_i + \sin(z_i),\ i = 1, 2, \cdots, n, b = Lu^*.$$

For both cases, we set the prescribed relative accuracy to be $\epsilon = 10^{-5}$ such that $\|\hat{u} - u^*\|_L \leq \epsilon \|b\|_2$. By Theorem 3.2.3, this accuracy can be achieved by imposing

a corresponding accuracy control on each level's linear system relative error (i.e., $\text{err}_A^{(K)}$ and $\text{err}_B$ ) and localization error ($\text{err}_{loc}$). In practice, instead of imposing a hard error control, we relax the localization error to $\varepsilon(\mathcal{P}^{(k)}, 1)$ (instead of $\epsilon$) in order to ensure sparsity, which is actually how we obtain the 4-level decomposition with localization. Such relaxed localization error can be fixed by doing a compensation correction at level-0, which takes the output of Algorithm 7 as an initialization to solve $Lu = b$. As shown in the gray columns "# Iteration" in Table 3.3 and Table 3.4, the number of iterations in the compensation calculation (which is the computation at Level 0) are fewer than 25 in both cases, which indicate that the localization error on each level is still small even when relaxed.

| | Scale | | Matrix | | # Iteration | | Main Cost | |
|---|---|---|---|---|---|---|---|---|
| Direct | $\lambda_{\max}^{-1}(L) \sim \lambda_{\min}^{-1}(L)$ | | $L$ | | 572 | | $7.32 \times 10^7$ | |
| Level | 4-level | 5-level | 4-lvl | 5-lvl | 4-lvl | 5-lvl | 4-lvl | 5-lvl |
| 0 | $10^{-5} \sim \lambda_{\max}^{-1}(L)$ | $10^{-5} \sim \lambda_{\max}^{-1}(L)$ | $\widetilde{B}^{(1)}$ | $\widetilde{B}^{(1)}$ | 22 | 22 | $2.15 \times 10^5$ | $2.15 \times 10^5$ |
| 1 | $10^{-5} \sim 10^{-4}$ | $10^{-5} \sim 10^{-4}$ | $\widetilde{B}^{(2)}$ | $\widetilde{B}^{(2)}$ | 11 | 11 | $4.31 \times 10^6$ | $4.31 \times 10^6$ |
| 2 | $10^{-4} \sim 10^{-3}$ | $10^{-4} \sim 3 \times 10^{-3}$ | $\widetilde{B}^{(3)}$ | $\widetilde{B}^{(3)}$ | 25 | 14 | $1.04 \times 10^7$ | $3.32 \times 10^6$ |
| | | $3 \times 10^{-3} \sim 10^{-3}$ | | $\widetilde{B}^{(4)}$ | | 14 | | $1.21 \times 10^6$ |
| 3 | $10^{-3} \sim 10^{-2}$ | $10^{-3} \sim 10^{-2}$ | $\widetilde{B}^{(4)}$ | $\widetilde{B}^{(5)}$ | 23 | 22 | $7.96 \times 10^5$ | $2.86 \times 10^5$ |
| 4 | $10^{-2} \sim \lambda_{\min}^{-1}(L)$ | $10^{-2} \sim \lambda_{\min}^{-1}(L)$ | $\widetilde{A}^{(4)}$ | $\widetilde{A}^{(5)}$ | 31 | 22 | $3.18 \times 10^4$ | $9.72 \times 10^3$ |
| 0 | - | - | L | L | 25 | 30 | $3.20 \times 10^6$ | $3.84 \times 10^6$ |
| Total | - | - | - | - | 137 | 135 | $1.90 \times 10^7$ | $1.31 \times 10^7$ |
| Parallel | - | - | - | - | - | - | $1.36 \times 10^7$ | $8.15 \times 10^6$ |

Table 3.3: Case 1: Performance of the 4-level and the 5-level decompositions.

| | Scale | | Matrix | | # Iteration | | Main Cost | |
|---|---|---|---|---|---|---|---|---|
| Direct | $\lambda_{\max}^{-1}(L) \sim \lambda_{\min}^{-1}(L)$ | | $L$ | | 586 | | $7.50 \times 10^7$ | |
| Level | 4-level | 5-level | 4-lvl | 5-lvl | 4-lvl | 5-lvl | 4-lvl | 5-lvl |
| 0 | $10^{-5} \sim \lambda_{\max}^{-1}(L)$ | $10^{-5} \sim \lambda_{\max}^{-1}(L)$ | $\widetilde{B}^{(1)}$ | $\widetilde{B}^{(1)}$ | 24 | 24 | $2.35 \times 10^5$ | $2.35 \times 10^5$ |
| 1 | $10^{-5} \sim 10^{-4}$ | $10^{-5} \sim 10^{-4}$ | $\widetilde{B}^{(2)}$ | $\widetilde{B}^{(2)}$ | 11 | 11 | $4.31 \times 10^6$ | $4.31 \times 10^6$ |
| 2 | $10^{-4} \sim 10^{-3}$ | $10^{-4} \sim 3 \times 10^{-3}$ | $\widetilde{B}^{(3)}$ | $\widetilde{B}^{(3)}$ | 25 | 14 | $1.04 \times 10^7$ | $3.32 \times 10^6$ |
| | | $3 \times 10^{-3} \sim 10^{-3}$ | | $\widetilde{B}^{(4)}$ | | 14 | | $1.21 \times 10^6$ |
| 3 | $10^{-3} \sim 10^{-2}$ | $10^{-3} \sim 10^{-2}$ | $\widetilde{B}^{(4)}$ | $\widetilde{B}^{(5)}$ | 23 | 23 | $7.96 \times 10^5$ | $2.99 \times 10^5$ |
| 4 | $10^{-2} \sim \lambda_{\min}^{-1}(L)$ | $10^{-2} \sim \lambda_{\min}^{-1}(L)$ | $\widetilde{A}^{(4)}$ | $\widetilde{A}^{(5)}$ | 30 | 22 | $3.08 \times 10^4$ | $9.72 \times 10^3$ |
| 0 | - | - | L | L | 16 | 23 | $2.05 \times 10^6$ | $2.94 \times 10^6$ |
| Total | - | - | - | - | 129 | 131 | $1.78 \times 10^7$ | $1.23 \times 10^7$ |
| Parallel | - | - | - | - | - | - | $1.24 \times 10^7$ | $7.25 \times 10^6$ |

Table 3.4: Case 2: Performance of the 4-level and the 5-level decompositions.

In particular, we use a preconditioned CG method to solve any involved linear systems $Au = b$ in Algorithm 7. The precondition matrix $D$ is chosen as the

diagonal part of $A$ and we take $\mathbf{0}$ (all-zeros vector) as initials if no preconditioning vector is provided. The main computational cost of a single use of the PCG method is measured by the product of the number of iterations and the NNZ entries of the matrix involved (See the gray column "Main Cost" in both Table 3.3 and Table 3.4).

In both cases, we can see that the total computational costs of our approach are obviously reduced compared to the direct use of PCG. Moreover, since the downward level-wise computation can be done in parallel, the effective computational costs of our approach are even less, which is the sum of the maximal cost among all levels and the cost of the compensation correction (See "Parallel" row in Table 3.3 and Table 3.4 respectively).

Further, from the results we can see that the costs among all levels in both cases are mainly concentrated on level 2, namely, the inverting of $\widetilde{B}^{(3)}$. 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 to the fact that $\widetilde{B}^{(3)}$ on level 2 has the largest complexity of all. Though in practice we do not have the information in Table 3.3 and Table 3.4, we may observe the dominance of time complexity 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 $\{\varepsilon(\mathcal{P}^{(k)}, 1)^2\}_{k=1}^{5} = \{0.00001, 0.0001, 0.0003, 0.001, 0.01\}$ (See Table 3.2 and Figure 3.4). From the "Main Cost 5-level" columns of Table 3.3 and Table 3.4), we can see that this simple improvement of further decomposition based on the feedback of computational results does reduce the computational cost.

*Chapter 4*

# HIERARCHICALLY PRECONDITIONED EIGENSOLVER

In this chapter, we develop our MMD framework into a hierarchically preconditioned eigensolver for large and sparse positive semidefinite matrices. The foundation of our eigensolver is the celebrated Implicit Restarted Lanczos Method (IRLM), which is integrated into a hierarchical structure consisting of two alternating processes: the level-wise spectrum extension process that finds new eigenpair candidates, and the cross-level spectrum refinement process that refines the computed eigenspace. In the extension process, we propose a spectral preserving preconditioner to guarantee the computational efficacy of each iteration. We apply the Orthogonal Iteration with Ritz Acceleration to design a efficient refinement process that converges exponentially fast with properly chosen parameters.

The layout of the rest of this chapter is as follows: We briefly review the Implicitly Restarted Lanczos Method in Section 4.1. Some spectrum error analysis and perturbation theories subject to our operator compression framework are discussed in Section 4.2. Theoretical developments and algorithms of the hierarchical spectrum extension process and the eigenpair refinement process are then proposed in Section 4.3 and Section 4.4 respectively. Combining these two procedures, we propose our hierarchical eigensolver in Section 4.5, where details of the choice of parameters are discussed. Section 4.6 is devoted to experimental results to justify the effectiveness of our proposed algorithm. In Section 4.7, we provide a quantitative numerical comparison with the conventional IRLM. The numerical results show that our proposed algorithm gives a promising results in terms of runtime complexity. In Section 4.8, we further compare our approach to a class of existing works on eigenspace approximation via the computation of compressed eigenmodes.

## 4.1 Implicitly Restarted Lanczos Method (IRLM)

The Arnoldi iteration is a widely used method to find eigenvalues of general asymmetric sparse matrices. It belongs to the family of Krylov subspace methods. In the symmetric case, we can further simplify it as the Lanczos iteration. A direct application of Lanczos iteration gives the largest eigenvalues of an operator by calculating the eigenvalues of its projection on a Krylov subspace. In each step the algorithm expands the Krylov subspace and finds an orthogonal basis of the space.

Namely, after $k$ steps, the factorization is

$$AV_k = V_k T_k + f_k e_k^T. \tag{4.1}$$

where we recall that $T_k$ is a tridiagonal matrix when $A$ is symmetric. Denote $(\theta, y)$ as an eigenpair of $T_k$. Let $x = V_k y$. Then we have

$$\|Ax - x\theta\|_2 = \|AV_k y - V_k y\theta\|_2 = \|f_k\|_2 |e_k^T y|. \tag{4.2}$$

Therefore $\theta$ is a good approximation of the eigenvalue of $A$ if and only if $\|f_k\|_2 |e_k^T y|$ is small. The latter is called the Ritz residual. An analogy to the power method shows that, to compute the largest $m$ eigenvalues, the convergence rate of the largest $m$ eigenvalues of $A$ is $(\lambda_{m+1}/\lambda_m)^k$ where $\lambda_i$ is the $i$th largest eigenvalue of $A$.

The direct Lanczos method is not practical due to the fact that $\|f_k\|_2$ rarely becomes small enough until the size of $T_k$ approaches that of $A$. An improvement is the IRLM [64, 109]. The IRLM employs the idea analogous to the implicitly shifted QR-iteration [39]. With this approach, the "unwanted" eigenvalues (in this case the leftmost ones) are shifted away implicitly in each round of implicit restart, and $T_k$ is kept with a small size equal to the number of desired eigenvalues. This is one of the state-of-the-art algorithms for large-scale partial eigen problems.

Yet, it is still complicated if we want to find the leftmost eigenvalues. One possible approach is to use a shifted IRLM. Namely, to find eigenvalues nearest to $\sigma$, we can replace $A$ with $(A - \sigma I)^{-1}$ as the target operator. By taking $\sigma = 0$ we get the eigenvalues with smallest magnitude. Such approach usually converges with a few iterations, but it requires solving $A^{-1}$ in every iteration. For large sparse problems, $A^{-1}$ is usually solved by the CG method. The complexity of CG is the complexity of matrix-vector product times the number of CG iterations. The former is equal to the number of nonzero entries of $A$ (denoted as $nnz(A)$), while the latter is controlled by the condition number $\kappa(A)$. Therefore, the total complexity of the shifted IRLM for solving $m_{tar}$ smallest eigenvalues is

$$O(R_{\text{IRLM}} \cdot m_{tar} \cdot nnz(A) \cdot \kappa(A)), \tag{4.3}$$

where $R_{\text{IRLM}}$ is the number of IRLM rounds. In the following, we will develop the extension-refinement algorithm to integrate the MMD framework with the shifted IRLM which gives considerable improvement in terms of iteration numbers of CG and PCG throughout the algorithm.

**Algorithm 8** Lanczos Iteration ($p$-step extension)

**Input:** $V, T, f$, target operator $op(\cdot)$, $p$.
**Output:** $V, T, f$.

$k$ = column number of $V$;
**for** $i = 1 : p$ **do**
  $\beta = \|f\|_2$;
  **if** $\beta < \epsilon$ **then**
    generate a new random $f$, $\beta = \|f\|_2$;
  **end if**
  $T \leftarrow \begin{pmatrix} T \\ \beta e_{k+i-1}^T \end{pmatrix}$, $v = f/\beta$, $V \leftarrow [V, v]$;
  $w = op(v)$;
  $h = V^T w$, $T \leftarrow [T, h]$;
  $f = w - Vh$;
  Re-orthogonalize to adjust $f$;
**end for**

**Algorithm 9** Inner Iteration of the IRLM

**Input:** $V, T, f, p$.
**Output:** $V, T, f$.

$k$ = column number of $V$;
Perform Algorithm 8 on $V, T$ and $f$ for $p$ steps;
Set $Q = I_{k+p}$ and $\{\sigma_j\}$ to be the $p$ smallest eigenvalues of $T$;
**for** $j = 1 : p$ **do**
  $T - \sigma_j I = Q_j R_j$;
  $T = Q_j^T T Q_j, Q \leftarrow QQ_j$;
**end for**
$f \leftarrow V \cdot Q(:, k+1) \cdot T(k+1, k) + f \cdot Q(k+p, k)$;
$V \leftarrow V \cdot Q(:, 1:k), T \leftarrow T(1:k, 1:k)$;

### 4.2 The Compressed Eigen Problem

In the previous section, we introduced an effective compression technique for a PD matrix $A$ subject to a prescribed compression error $\epsilon$. The compressed operator is also symmetric and positive definite. Therefore, by the well-known eigenvalue perturbation theory, we know that the eigenpairs of the compressed operator can be used as good approximations for the eigenpairs of the original matrix. In particular, we have the following estimate:

**Lemma 4.2.1.** *Let $\Theta = \Psi(\Psi^T A \Psi)^{-1}\Psi^T$ be a rank-N compressed approximation of $A^{-1}$ introduced in Theorem 2.2.1 such that $\|A^{-1} - \Theta\|_2 \leq \varepsilon$. Let $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_n > 0$ be the eigenvalues of $A^{-1}$ in a descending order, and $\tilde{\mu}_1 \geq \tilde{\mu}_2 \geq \cdots \geq \tilde{\mu}_N >$*

0 *be the nonzero eigenvalues of* $\Theta$ *in a descending order. Then we have*

$$|\mu_i - \tilde{\mu}_i| \le \varepsilon, \quad 1 \le i \le N; \qquad \mu_i \le \varepsilon, \quad N < i \le n.$$

*Moreover, let* $\tilde{v}_i$, $i = 1, \cdots, N$, *be the corresponding normalized eigenvectors of* $\Theta$ *such that* $\Theta\tilde{v}_i = \tilde{\mu}_i\tilde{v}_i$, *then we have*

$$\|A^{-1}\tilde{v}_i - \mu_i\tilde{v}_i\|_2 \le 2\varepsilon, \quad 1 \le i \le N.$$

Since the nonzero eigenvalues of $\Theta$ and the corresponding eigenvectors actually result from the nonsingular stiffness matrix $A_{st} = \Psi^T A\Psi$, we will call these eigenpairs the **essential eigenpairs** of $\Theta$ in what follows. We will also need the following lemma for developing our algorithms.

**Lemma 4.2.2.** *Let* $(\tilde{\mu}_i, \tilde{v}_i)$, $i = 1, \cdots, N$, *be the* $N$ *essential eigenpairs of* $\Theta$ *given in Lemma 4.2.1.*

*(i) Let* $w_i = \Psi^T\tilde{v}_i$, *then*

$$\Psi^T\Psi A_{st}^{-1}w_i = \tilde{\mu}_i w_i, \quad 1 \le i \le N.$$

*(ii) Let* $z_i = \Psi^\dagger\tilde{v}_i = (\Psi^T\Psi)^{-1}\Psi^T\tilde{v}_i$, *then*

$$A_{st}^{-1}\Psi^T\Psi z_i = \tilde{\mu}_i z_i, \quad 1 \le i \le N,$$

*where* $A_{st} = \Psi^T A\Psi$ *is the stiffness matrix. Conversely, if either (i) or (ii) is true, then* $(\tilde{\mu}_i, \tilde{v}_i)$, $i = 1, \cdots, N$, *are eigenpairs of* $\Theta$.

Similar to Lemma 4.2.1, we have the following estimates for multiresolution decomposition.

**Lemma 4.2.3.** *Given an integer* $K$, *let* $\Theta^{(k)} = \mathbf{\Psi}^{(k)}\big((\mathbf{\Psi}^{(k)})^T A\mathbf{\Psi}^{(k)}\big)^{-1}(\mathbf{\Psi}^{(k)})^T$, $k = 1, 2, \cdots, K$, *with* $\mathbf{\Psi}^{(k)}$ *given in* (3.9). *Write* $A^{-1} = \Theta^{(0)}$. *Let* $(\mu_i^{(k)}, v_i^{(k)})$, $i = 1, 2, \cdots, N^{(k)}$, *be the essential eigenpairs of* $\Theta^{(k)}$ *where* $\mu_1^{(k)} \ge \mu_2^{(k)} \ge \cdots \ge \mu_{N^{(k)}}^{(k)} > 0$. *Then for any* $0 \le k' < k \le K$, *we have*

$$|\mu_i^{(k')} - \mu_i^{(k)}| \le \varepsilon_k, \quad 1 \le i \le N^{(k)}; \qquad |\mu_i^{(k')}| \le \varepsilon_k, \quad N^{(k)} < i \le N^{(k')},$$

*and*

$$\|\Theta^{(k')}v_i^{(k)} - \mu_i^{(k')}v_i^{(k)}\|_2 \le 2\varepsilon_k, \quad 1 \le i \le N^{(k)}.$$

*Proof.* By Theorem 3.3.1 we have that $\|\Theta^{(0)} - \Theta^{(k)}\|_2 = \|A^{-1} - \Theta^{(k)}\|_2 \leq \varepsilon_k$, $k = 1, 2, \cdots, K$. From the definition of $\Theta^{(k)}$ and the decomposition (3.10), one can easily check that

$$A^{-1} = \Theta^{(0)} \succeq \Theta^{(1)} \succeq \cdots \succeq \Theta^{(K-1)} \succeq \Theta^{(K)}.$$

Then the results follow immediately. □

**On Compressed Eigenproblems**

Recall that the efficiency of constructing the compressed operator we propose relies on the exponential decay property of the basis $\Psi$. This spacial exponential decay feature allows us to localize $\Psi$ and to construct sparse stiffness matrix $A_{st} = \Psi^T A \Psi$ without compromising compression accuracy $\varepsilon$ in $O(nnz(A) \cdot (\log(\frac{1}{\varepsilon}) + \log n)^c)$ time. In fact, the problem of using spatially localized/compact basis to compress high-dimensional operator and to approximate eigenspace of smallest eigenvalues has long been studied in different ways. A representative pioneer work is the method of compressed modes proposed by Ozoliņš et al. [90], intended originally for Schrödinger's equation in quantum physics. By adding a $L_1$ regularization to the variational form of an eigenproblem, they obtained spatially compressed basis modes that well span the desired eigenspace. Though the way they obtain sparsity is quite different from what we do, both methods obtain interestingly similar results for some model problems. It can be inspiring to make comparison between their method and ours, so that readers can have better understanding of our approach. We leave the detailed comparison to Section 4.2.

## 4.3 Hierarchical Spectrum Completion

Now that we have a sequence of compressed approximations, we next seek to use this decomposition to compute the dominant spectrum of $A^{-1}$ down to a prescribed value in a hierarchical manner. In particular, we propose to decompose the target spectrum into several segments of different scales, and then allocate the computation of each segment to a certain level of the compressing sequence so that the problem on each level is well-conditioned.

To implement this idea, we first go back to the one-level compression settings. Suppose that we have accurately obtained the first $m$ essential eigenpairs $(\mu_i, v_i)$, $i = 1, \cdots, m$, of $\Theta = \Psi(\Psi^T A \Psi)^{-1}\Psi^T = \Psi A_{st}^{-1}\Psi^T$, and our aim is to compute the following $m_{tar} - m$ eigenpairs (namely extend to the first $m_{tar}$ eigenpairs) using the Lanczos method. Define $V_m = \text{span}\{v_i : 1 \leq i \leq m\}$ and $V_{m^+} = \text{span}\{v_i : m < i \leq$

$N\} = V_m^{\perp} \cap \text{span}\{\Psi\}$. Then to perform the Lanczos method to compute the next segment of eigenpairs of $\Theta$, we need to repeatedly apply the operator $\Psi A_{st}^{-1}\Psi^T$ to vectors in $V_{m^+}$, which requires to compute $A_{st}^{-1}w$ for $w \in W_{m^+} = \Psi^T(V_{m^+})$.

Ideally we want the computation of the following $m_{tar} - m$ eigenpairs to be restricted to a problem with bounded spectrum width that is proportional to $\mu_m/\mu_{m_{tar}}$. This is possible since we assume that we have accurately obtained the span space $V_m$ of the first $m$ eigenvectors, and thus we can consider our problem in the reduced space orthogonal to $V_m$. In this case, the CG method will be efficient for computing inverse matrix operations.

**Definition 4.3.1.** *Let $A$ be a symmetric, positive definite matrix, and $V$ be an invariant subspace of $A$. We define the condition number of $A$ with respect to $V$ as*

$$\kappa(A, V) = \frac{\lambda_{max}(A, V)}{\lambda_{min}(A, V)},$$

*where*

$$\lambda_{max}(A, V) = \max_{v \in V, v \neq 0} \frac{v^T A v}{v^T v}, \qquad \lambda_{min}(A, V) = \min_{v \in V, v \neq 0} \frac{v^T A v}{v^T v}.$$

**Theorem 4.3.2.** *Let $A$ be a symmetric, positive definite matrix, and $V$ be an invariant subspace of $A$. When using the conjugate gradient method to solve $Ax = b$ with initial guess $x_0$ such that $r_0 = b - Ax_0 \in V$, we have the following estimate*

$$\|x_k - x_*\|_A \leq 2\Big(\frac{\sqrt{\kappa(A, V)} - 1}{\sqrt{\kappa(A, V)} + 1}\Big)^k \|x_0 - x_*\|_A,$$

*and*

$$\|x_k - x_*\|_2 \leq 2\sqrt{\kappa(A, V)}\Big(\frac{\sqrt{\kappa(A, V)} - 1}{\sqrt{\kappa(A, V)} + 1}\Big)^k \|x_0 - x_*\|_2,$$

*where $x_*$ is the exact solution, and $x_k \in x_* + V$ is the solution at the $k^{th}$ step of CG iteration. Thus it takes $k = O(\kappa(A, V) \cdot \log \frac{1}{\epsilon})$ steps (or $k = O\big(\kappa(A, V) \cdot (\log \kappa(A, V) + \log \frac{1}{\epsilon})\big)$ steps) to obtain a solution subject to relative error $\epsilon$ in the energy norm (or $l_2$ norm).*

*Proof.* We only need to notice that the $k$-order Krylov subspace $\mathcal{K}(A, r_0, k)$ generated by $A$ and $r_0$ satisfies

$$\mathcal{K}(A, r_0, k) \subset V, \quad \forall k \in \mathbb{Z}.$$

□

Notice that, for any $i = m+1, \cdots, N$, though $v_i \in V_{m^+}$ is an eigenvector of $\Theta = \Psi A_{st}^{-1}\Psi^T$, $w_i = \Psi^T v_i$ is not an eigenvector of $A_{st}^{-1}$(but an eigenvector of $\Psi^T\Psi A_{st}^{-1}$) since we do not require $\Psi$ to be orthonormal. Therefore the space $W_{m^+}$ is not an invariant space of $A_{st}$, and if we directly use the CG method to solve $A_{st}x = w$, the convergence rate will depend on $\kappa(A_{st}) = \lambda_{max}(A_{st})\lambda_{min}(A_{st})^{-1}$ , instead of $\lambda_{max}(A_{st})\mu_m$ as intended. Though we bound $\lambda_{max}(A_{st})$ from above by $\delta(\mathcal{P})$ and $\lambda_{min}(A_{st})$ from below by $\lambda_{min}(A)$ (See Theorem 2.2.25), $\kappa(A_{st})$ can be still large since we prescribe a bounded compression rate in practice to ensure the efficiency of the compression algorithm.

Therefore, we need to find a proper invariant space, so that we can make use of the knowledge of the space $V_m$ and restrict the computation of $A_{st}^{-1}w$ to a problem of narrower spectrum.

**Lemma 4.3.3.** *Let $(\mu_i, v_i)$, $i = 1, \cdots, N$, be the essential eigenpairs of $\Theta = \Psi(\Psi^T A\Psi)^{-1}\Psi^T = \Psi A_{st}^{-1}\Psi^T$, such that $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_N > 0$. Let $(\Psi^T\Psi)^{\frac{1}{2}}$ be the square root of the symmetric, positive definite matrix $\Psi^T\Psi$. Then $(\mu_i, z_i)$, $i = 1, \cdots, N$, are all eigenpairs of $(\Psi^T\Psi)^{\frac{1}{2}}A_{st}^{-1}(\Psi^T\Psi)^{\frac{1}{2}}$, where*

$$z_i = (\Psi^T\Psi)^{-\frac{1}{2}}\Psi^T v_i, \quad 1 \leq i \leq N.$$

*Moreover, for any subset $S \subset \{1, 2, \cdots, N\}$, and $Z_S = \text{span}\{z_i : i \in S\}$, we have*

$$\mathcal{K}(A_\Psi, z, k) \subset Z_S, \quad \forall z \in Z_S, \ \forall k \in \mathbb{Z},$$

*where $A_\Psi = (\Psi^T\Psi)^{-\frac{1}{2}}A_{st}(\Psi^T\Psi)^{-\frac{1}{2}}$.*

**Lemma 4.3.4.** *Let $\Psi$ be given in Construction 2.2.9, then we have*

$$\lambda_{min}(\Psi^T\Psi) \geq 1, \quad \lambda_{max}(\Psi^T\Psi) \leq 1 + \varepsilon(\mathcal{P})\delta(\mathcal{P}),$$

*and thus*

$$\kappa(\Psi^T\Psi) \leq 1 + \varepsilon(\mathcal{P})\delta(\mathcal{P}).$$

*Proof.* Let $U$ be the orthogonal complement basis of $\Phi$ as given in (3.2), so $[\Phi, U]$ is an orthonormal basis of $\mathbb{R}^n$, and we have $\Phi\Phi^T + UU^T = I_n$. Since $\Phi^T\Psi = \Phi^T A^{-1}\Phi(\Phi^T A^{-1}\Phi)^{-1} = I_N$, we have

$$\Psi^T\Psi = \Psi^T\Phi\Phi^T\Psi + \Psi^T UU^T\Psi = I_N + \Psi^T UU^T\Psi.$$

We then immediately obtain $\Psi^T\Psi \geq I_N$, and thus $\lambda_{min}(\Psi^T\Psi) \geq 1$. To obtain an upper bound of $\lambda_{max}(\Psi^T\Psi)$, we notice that from the construction of $\Phi$ we have

$$\|x - P_\Phi x\|_2^2 \leq \varepsilon(\mathcal{P}) x^T A x, \quad \forall x \in \mathbb{R}^n \quad \Longrightarrow \quad (I_n - P_\Phi)^2 \leq \varepsilon(\mathcal{P}) A,$$

where $P_\Phi = \Phi\Phi^T$ denotes the orthogonal projection into span$\{\Phi\}$. Since $\Phi\Phi^T + UU^T = I_n$, we have

$$UU^T = I_n - \Phi\Phi^T = (I_n - \Phi\Phi^T)^2 \leq \varepsilon(\mathcal{P}) A.$$

Therefore we have

$$\Psi^T\Psi = I_N + \Psi^T UU^T \Psi \leq I_N + \varepsilon(\mathcal{P})\Psi^T A \Psi = I_N + \varepsilon(\mathcal{P}) A_{st},$$

and by Theorem 2.2.25 we obtain

$$\lambda_{max}(\Psi^T\Psi) \leq 1 + \varepsilon(\mathcal{P})\lambda_{max}(A_{st}) \leq 1 + \varepsilon(\mathcal{P})\delta(\mathcal{P}).$$

□

**Theorem 4.3.5.** *Let $A_\Psi$ and $(\mu_i, z_i)$ be defined as in Lemma 4.3.3. Let $Z_{m^+} = \text{span}\{z_i : m < i \leq N\}$, then $Z_{m^+}$ is an invariant space of $A_\Psi$, and we have*

$$\kappa(A_\Psi, Z_{m^+}) \leq \mu_{m+1}\delta(\mathcal{P}).$$

*Proof.* By Lemma 4.3.3 and Lemma 4.3.4, we have

$$\begin{aligned}\lambda_{max}(A_\Psi, Z_{m^+}) \leq \lambda_{max}(A_\Psi) &= \|(\Psi^T\Psi)^{-\frac{1}{2}} A_{st} (\Psi^T\Psi)^{-\frac{1}{2}}\|_2 \\ &\leq \|A_{st}\|_2 \|(\Psi^T\Psi)^{-1}\|_2 \leq \delta(\mathcal{P}).\end{aligned}$$

And by the definition of $Z_{m^+}$, we have

$$\lambda_{min}(A_\Psi, Z_{m^+}) = \frac{1}{\lambda_{max}(A_\Psi^{-1}, Z_{m^+})} = \frac{1}{\lambda_{max}((\Psi^T\Psi)^{\frac{1}{2}} A_{st}^{-1} (\Psi^T\Psi)^{\frac{1}{2}}, Z_{m^+})} = \frac{1}{\mu_{m+1}}.$$

□

Inspired by Lemma 4.3.4 and Theorem 4.3.5, we now consider to solve $A_{st}x = w$ efficiently for $w \in W_{m^+} = \Psi^T(V_{m^+}) = (\Psi^T\Psi)^{\frac{1}{2}}(Z_{m^+})$ by making use of the controlled condition number $\kappa(A_\Psi, Z_{m^+})$ and $\kappa(\Psi^T\Psi)$. Theoretically, we can compute $x = A_{st}^{-1} w$ by the following steps:

(i) Compute $b = (\Psi^T\Psi)^{-\frac{1}{2}} w \in Z_{m^+}$;

(ii) Use the CG method to compute $y = A_\Psi^{-1} b$ with initial guess $y_0$ such that $b - A_\Psi y_0 \in Z_{m^+}$;

(iii) Compute $x = (\Psi^T\Psi)^{-\frac{1}{2}} y$.

Notice that this procedure is exactly solving $A_{st}x = w$ using the preconditioned CG method with preconditioner $\Psi^T\Psi$, which only involves applying $A_{st}$ and $(\Psi^T\Psi)^{-1}$ to vectors, but still enjoys the good conditioning property of $A_\Psi$ restricted to $Z_{m^+}$. Therefore we have the following estimate:

**Corollary 4.3.6.** *Consider using the PCG method to solve $A_{st}x = w$ for $w \in W_{m^+}$ with preconditioner $\Psi^T\Psi$ and initial guess $x_0$ such that $r_0 = w - A_{st}x_0 \in W_{m^+}$. Let $x_*$ be the exact solution, and $x_k$ be the solution at the $k^{th}$ step of the PCG iteration. Then we have*

$$\|x_k - x_*\|_{A_{st}} \leq 2\Big(\frac{\sqrt{\kappa(A_\Psi, Z_{m^+})} - 1}{\sqrt{\kappa(A_\Psi, Z_{m^+})} + 1}\Big)^k \|x_0 - x_*\|_{A_{st}},$$

*and*

$$\|x_k - x_*\|_2 \leq 2\sqrt{\kappa(\Psi^T\Psi)\kappa(A_\Psi, Z_{m^+})}\Big(\frac{\sqrt{\kappa(A_\Psi, Z_{m^+})} - 1}{\sqrt{\kappa(A_\Psi, Z_{m^+})} + 1}\Big)^k \|x_0 - x_*\|_2.$$

*Proof.* Let $y_k = (\Psi^T\Psi)^{\frac{1}{2}} x_k$ and $y_* = (\Psi^T\Psi)^{\frac{1}{2}} x_*$, then we have

$$\|y_k - y_*\|_2^2 = (x_k - x_*)^T \Psi^T\Psi (x_k - x_*),$$

and

$$\|y_k - y_*\|_{A_\Psi}^2 = (y_k - y_*)^T A_\Psi (y_k - y_*) = \|x_k - x_*\|_{A_{st}}^2.$$

Noticing that $(\Psi^T\Psi)^{-\frac{1}{2}} r_0 \in Z_{m^+}$ and $\mathcal{K}(A_\Psi, (\Psi^T\Psi)^{-\frac{1}{2}} r_0, k) \subset Z_{m^+}$ $\forall k$, the results follow from Theorem 4.3.2. □

By Corollary 4.3.6, to compute a solution of $A_{st}x = w$ subject to a relative error $\epsilon$ in the $A_{st}$-norm, the number of needed PCG iterations is

$$O\big(\kappa(A_\Psi, Z_{m^+}) \cdot \log\frac{1}{\epsilon}\big) = O\big(\mu_{m+1}\delta(\mathcal{P}) \cdot \log\frac{1}{\epsilon}\big).$$

This is also an estimate of the number of needed PCG iterations for a relative error $\epsilon$ in the $l_2$-norm, if we assume that $\kappa(\Psi^T\Psi), \kappa(A_\Psi, Z_{m^+}) \leq \frac{1}{\epsilon}$.

In what follows we will denote $M = \Psi^T\Psi$. Notice that the nonzero entries of $M$ are due to the overlapping support of column basis vectors of $\Psi$, while the nonzero entries of $A_{st} = \Psi^T A\Psi$ are results of interactions between column basis vectors of $\Psi$ with respect to $A$. Thus we can reasonably assume that $nnz(M) \leq nnz(A_{st})$. Suppose that in each iteration of the whole PCG procedure, we also use the CG method to compute $M^{-1}b$ for some $b$ subject to a relatively higher precision $\hat{\epsilon}$, which requires a cost of $O(nnz(M) \cdot \kappa(M) \cdot \log \frac{1}{\hat{\epsilon}})$. In practice it is sufficient to take $\hat{\epsilon}$ smaller than but comparable to $\epsilon$ (e.g. $\hat{\epsilon} = 0.1\epsilon$), so $\log(\frac{1}{\hat{\epsilon}}) = O(\log \frac{1}{\epsilon})$. By Lemma 4.3.4 we have $\kappa(M) = O(\varepsilon(\mathcal{P})\delta(\mathcal{P}))$. Then the computational complexity of each single iteration can be bounded by

$$O\big(nnz(A_{st})\big) + O\big(nnz(M) \cdot \kappa(M) \cdot \log \frac{1}{\epsilon}\big) = O\big(nnz(A_{st}) \cdot \varepsilon(\mathcal{P})\delta(\mathcal{P}) \cdot \log \frac{1}{\epsilon}\big),$$

and the total cost of computing a solution of $A_{st}x = w$ subject to a relative error $\epsilon$ is

$$O\big(\mu_{m+1}\delta(\mathcal{P}) \cdot nnz(A_{st}) \cdot \varepsilon(\mathcal{P})\delta(\mathcal{P}) \cdot (\log \frac{1}{\epsilon})^2\big). \tag{4.4}$$

We remark that when the original size of $A \in \mathbb{R}^{n\times n}$ is large, the eigenvectors $V$ are long and dense. It would be expensive to compute inner products with these long vectors over and over again. In fact, in the previous discussions the operator $\Theta = \Psi A_{st}^{-1}\Psi^T$(of the same size as $A$) and the eigenvectors $V$ are only for purpose of analysis use to explain the idea of our method. In practical, for a long vector $v = \Psi\hat{v}$, we do not need to keep track of the whole vector, but only need to store its much shorter coefficients $\hat{v}$ of compressed dimension $N$ instead. When we compute $v_2 = \Theta v_1 = \Psi A_{st}^{-1}\Psi^T v_1$, it is equivalent to computing $\hat{v}_2 = A_{st}^{-1}M\hat{v}_1$, where $v_j = \Psi\hat{v}_j,\ j = 1, 2$, and $M = \Psi^T\Psi$. One can check that the analysis presented above still applies. So in the implementation of our method, we only deal with operator $A_{st}^{-1}M$ and short vectors $\widehat{V}$, and the long eigenvectors $V$ and $\Psi$ will not appear until in the very end when we recover $V = \Psi\widehat{V}$. We remark that since the eigenvectors of $\Theta$ are orthogonal, their coefficient vectors $\widehat{V}$ are $M$-orthogonal, i.e. $\widehat{V}^T M\widehat{V} = I$. We use $\|x\|_M$ to denote the norm $\sqrt{x^T Mx}$.

Recall that in the Lanczos method with respect to operator $\Theta$, the upper-Hessenberg matrix $T$ in the Arnoldi relation

$$\Theta V = VT + fe^T$$

is indeed tridiagonal, since $\Theta$ is symmetric, and $V^T[V, f] = [I, \mathbf{0}]$. This upper-Hessenberg matrix $T$ being tridiagonal is the reason why the implicit restarting process(Algorithm 9) is efficient. Now since we are actually dealing with the operator $A_{st}^{-1}M$ and the coefficient vectors $\widehat{V}^T = M^{-1}\Psi^T V$, the Arnoldi relation becomes

$$A_{st}^{-1}M\widehat{V} = \widehat{V}T + \hat{f}e^T,$$

where $\hat{f} = M^{-1}\Psi^T f$. So as long as we keep $\widehat{V}$ $M$-orthogonal and $\hat{f}$ $M$-orthogonal to $\widehat{V}$, $T$ will still be tridiagonal since

$$T = \widehat{V}^T M\widehat{V}T = \widehat{V}^T M(\widehat{V}T + \hat{f}e^T) = \widehat{V}^T M A_{st}^{-1} M\widehat{V}$$

is symmetric. We therefore modified Algorithm 8 to Algorithm 10 to take $M$-orthogonality into consideration.

Summarizing the analysis above, we propose Algorithm 11 for extending a given collection of eigenpairs using the Lanczos type method. The operator $OP(\,\cdot\,; A_{st}, M, \epsilon_{op})$ exploits our key idea that uses $M = \Psi^T\Psi$ as the preconditioner to effectively reduce the number of PCG iterations in every operation of $A_{st}^{-1}M$. For convenience, we will use "$x = pcg(A, b, M, x_0, \epsilon)$" to represent the operation of computing $x = A^{-1}b$ using the PCG method with preconditioner $M$ and initial guess $x_0$, subject to relative error $\epsilon$. "$x = pcg(A, b, -, x_0, \epsilon)$" means no preconditioner is used (i.e. the normal CG method), and "$x = pcg(A, b, M, -, \epsilon)$" means an all zero vector is used as the initial guess.

**Algorithm 10** General Lanczos Iteration ($p$-step extension)

**Input:** $\widehat{V}, T, \hat{f}$, target operator $op(\cdot)$, $p$, inner product matrix $M$
**Output:** $\widehat{V}, T, \hat{f}$

1: $k$ = column number of $\widehat{V}$;
2: **for** $i = 1 : p$ **do**
3: $\quad\beta = \|\hat{f}\|_M$;
4: $\quad$**if** $\beta < \epsilon$ **then**
5: $\quad\quad$generate a new random $\hat{f}$, $\beta = \|\hat{f}\|_M$;
6: $\quad$**end if**
7: $\quad T \leftarrow \begin{pmatrix} T \\ \beta e_{k+i-1}^T \end{pmatrix}$, $\quad \hat{v} = \hat{f}/\beta$, $\quad \widehat{V} \leftarrow [\widehat{V}, \hat{v}]$;
8: $\quad w = op(\hat{v})$;
9: $\quad h = \widehat{V}^T M w$, $\quad T \leftarrow [T, h]$;
10: $\quad \hat{f} = w - \widehat{V}h$;
11: $\quad$Re-orthogonalize to adjust $f$(with respect to $M$-orthogonality);
12: **end for**

Function $y$ = **Operator** $OP(x; A_{st}, M, \epsilon_{op})$

$w = Mx$;
$y = pcg(A_{st}, w, M, -, \epsilon_{op})$;

**Algorithm 11** Eigenpair Extension

**Input:** $\widehat{V}_{ini}$, $D_{ini}$, $OP(\,\cdot\,; A_{st}, M, \epsilon_{op})$, target number $m_{tar}$, prescribed accuracy $\epsilon$, eigenvalue threshold $\mu$, searching step $d$.
**Output:** $\widehat{V}_{ex}$, $D_{ex}$.

Generate random initial vector $\widehat{V} = \hat{v}$ that is $M$-orthogonal to $\widehat{V}_{ini}$;
**repeat**
  perform $d$ steps of general Lanczos iteration (Algorithm 10) with operator $OP$ to extend $\widehat{V}, T$;
  **while** Lanczos residual $> \epsilon$, **do**
    Perform $c \cdot d$ steps of shifts to restart Lanczos (Algorithm 9) and renew $\widehat{V}, T$;
  **end while**
  Find the smallest eigenvalue of $T$ as $\hat{\mu}$;
**until** $\hat{\mu} < \mu$ or $\dim(\widehat{V}) \geq m_{tar} - \dim(\widehat{V}_{ini})$.
$m_{new} = \dim(\widehat{V})$;
**while** Lanczos residual $> \epsilon$, **do**
  Perform $c \cdot m_{new}$ steps of shifts to restart Lanczos (Algorithm 9) and renew $\widehat{V}, T$;
**end while**
$PSP^T = T$ (Schur Decomposition);
$\widehat{V}_{ex} = [\widehat{V}_{ini}, \widehat{V}P]$, $\quad D_{ex} = \begin{bmatrix} D_{ini} & \\ & S \end{bmatrix}$;

Given an existing eigenspace $V_{ini} = \Psi\widehat{V}_{ini}$, Algorithm 11 basically uses the Lanczos method to find the following eigenpairs of $\Theta$ in the space $V_{ini}^{\perp}$. Notice that the output $\widehat{V}_{ex}$ gives the coefficients of the desired eigenvectors $V_{ex}$ in the basis $\Psi$. However, different from the classical Lanczos method, we do not prescribe a specific number for the output eigenpairs. Instead, we set a threshold $\mu$ to bound the last output eigenvalue. As we will develop our idea into a multi-level algorithm that pursues a number of target eigenpairs hierarchically, the output of the current level will be used to generate the initial eigenspace for the higher level. Therefore, the purpose of setting a threshold $\mu$ on the current level is to bound the restricted condition number on the higher level, as the initial eigenspace $V_{ini}$ from the lower level helps to bound the restricted condition number on the current level.

The choice of the threshold $\mu$ will be discussed in detail after we introduce the refinement procedure. Here, to develop a hierarchical spectrum completion method

using the analysis above, we state the hierarchical versions of Lemma 4.3.4 and Theorem 4.3.5.

**Lemma 4.3.7.** *Let $\mathbf{\Psi}^{(k)}$ be given in* (3.9)*, and $M^{(k)} = (\mathbf{\Psi}^{(k)})^T\mathbf{\Psi}^{(k)}$. Then we have*

$$\lambda_{min}(M^{(k)}) \geq 1, \quad \lambda_{max}(M^{(k)}) \leq 1 + \varepsilon_k\delta_k,$$

*and thus*

$$\kappa(M^{(k)}) \leq 1 + \varepsilon_k\delta_k.$$

*Proof.* The proof is similar to the proof of Lemma 4.3.4. Let $\boldsymbol{U}^{(k)} = (\mathbf{\Phi}^{(k)})^{\perp}$ be the orthogonal complement basis of $\mathbf{\Phi}^{(k)}$. According to Theorem 3.3.1, we have

$$\|x - P_{\mathbf{\Phi}^{(k)}}x\|_2^2 \leq \varepsilon_k\|x\|_A^2,$$

which implies that

$$\boldsymbol{U}^{(k)}(\boldsymbol{U}^{(k)})^T = (I_n - \mathbf{\Phi}^{(k)}(\mathbf{\Phi}^{(k)})^T) \leq \varepsilon_k A.$$

Notice that $(\mathbf{\Phi}^{(k)})^T\mathbf{\Psi}^{(k)} = I_{N^{(k)}}$, $\mathbf{\Phi}^{(k)}(\mathbf{\Phi}^{(k)})^T + \boldsymbol{U}^{(k)}(\boldsymbol{U}^{(k)})^T = I_n$, we thus have

$$\begin{aligned} M^{(k)} &= (\mathbf{\Psi}^{(k)})^T\mathbf{\Phi}^{(k)}(\mathbf{\Phi}^{(k)})^T\mathbf{\Psi}^{(k)} + (\mathbf{\Psi}^{(k)})^T\boldsymbol{U}^{(k)}(\boldsymbol{U}^{(k)})^T\mathbf{\Psi}^{(k)} \\ &= I_{N^{(k)}} + (\mathbf{\Psi}^{(k)})^T\boldsymbol{U}^{(k)}(\boldsymbol{U}^{(k)})^T\mathbf{\Psi}^{(k)}, \end{aligned}$$

$$\Longrightarrow \qquad I_{N^{(k)}} \leq M^{(k)} \leq I_{N^{(k)}} + \varepsilon_k(\mathbf{\Psi}^{(k)})^T A\mathbf{\Psi}^{(k)} = I_{N^{(k)}} + \varepsilon_k A^{(k)}.$$

Therefore we have $\lambda_{min}(M^{(k)}) \geq 1$, and by Corollary 3.1.4 we have

$$\lambda_{max}(M^{(k)}) \leq 1 + \varepsilon_k\lambda_{max}(A^{(k)}) \leq 1 + \varepsilon_k\delta_k.$$

□

**Theorem 4.3.8.** *Let $A^{(k)}$ and $\mathbf{\Psi}^{(k)}$ be given in* (3.9)*, and $M^{(k)} = (\mathbf{\Psi}^{(k)})^T\mathbf{\Psi}^{(k)}$. Let $(\mu_i^{(k)}, v_i^{(k)})$, $i = 1, \cdots, N^{(k)}$, be the essential eigenpairs of $\Theta^{(k)} = \mathbf{\Psi}^{(k)}(A^{(k)})^{-1}(\mathbf{\Psi}^{(k)})^T$. Define*

$$z_i^{(k)} = (M^{(k)})^{-\frac{1}{2}}(\mathbf{\Psi}^{(k)})^T v_i^{(k)}, \quad 1 \leq i \leq N^{(k)}.$$

*Given an integer $m_k$, let $Z_{m_k^+}^{(k)} = \text{span}\{z_i^{(k)} : m_k < i \leq N^{(k)}\}$, then $Z_{m_k^+}^{(k)}$ is an invariant space of $A_{\mathbf{\Psi}}^{(k)} = (M^{(k)})^{-\frac{1}{2}}A^{(k)}(M^{(k)})^{-\frac{1}{2}}$, and we have*

$$\kappa(A_{\mathbf{\Psi}}^{(k)}, Z_{m^+}^{(k)}) \leq \mu_{m_k+1}^{(k)}\delta_k.$$

*Moreover, consider using the PCG method to solve* $A^{(k)}x = w$ *for* $w \in W^{(k)}_{m_k^+}$ *with preconditioner* $M^{(k)}$ *and initial guess* $x_0$ *such that* $r_0 = w - A^{(k)}x_0 \in W^{(k)}_{m_k^+}$*, where* $W^{(k)}_{m_k^+} = \mathrm{span}\{(\mathbf{\Psi}^{(k)})^T v_i^{(k)} : m_k < i \le N^{(k)}\}$*. Let* $x_*$ *be the exact solution, and* $x_t$ *be the solution at the* $t^{th}$ *step of the PCG iteration. Then we have*

$$\|x_t - x_*\|_{A^{(k)}} \le 2\Big(\frac{\sqrt{\mu^{(k)}_{m_k+1}\delta_k} - 1}{\sqrt{\mu^{(k)}_{m_k+1}\delta_k} + 1}\Big)^t \|x_0 - x_*\|_{A^{(k)}},$$

*and*

$$\|x_t - x_*\|_2 \le 2\sqrt{\varepsilon_k \mu^{(k)}_{m_k+1}\delta_k^2}\Big(\frac{\sqrt{\mu^{(k)}_{m_k+1}\delta_k} - 1}{\sqrt{\mu^{(k)}_{m_k+1}\delta_k} + 1}\Big)^t \|x_0 - x_*\|_2.$$

Recall that we will use the CG method to implement Lanczos iteration on each level $k$ to complete the target spectrum. To ensure the efficiency of the CG method, namely to bound the restricted condition number $\kappa(A^{(k)}_{\mathbf{\Psi}}, Z^{(k)}_{m^+})$ on each level, we need a priori knowledge of the spectrum $\{(\mu_i^{(k)}, v_i^{(k)}) : 1 \le i \le m_k\}$ such that $\mu^{(k)}_{m_k+1}\delta_k$ is uniformly bounded. This given spectrum should be inductively computed on the lower level $k+1$. But notice that there is a compression error between each two neighbor levels, which will compromise the orthogonality and thus the theoretical bound for restricted condition number, if we directly use the spectrum of the lower level as a priori spectrum of the current level. Therefore we introduce a refinement method in Section 4.4 to overcome this difficulty.

### 4.4 Cross-Level Refinement of Eigenspace

In the previous section we have established a one-level spectrum extension method, given that a partial accurate spectrum is provided. To develop this method into an inductive hierarchical spectrum completion procedure, a natural idea is to use the spectrum computed at the lower level as the initial spectrum to be used in the higher level. However, such initial spectrum is not actually good enough since there is a compression error between each two neighboring levels. Thus we need to use a compatible refinement technique to refine the initial spectrum.

Now consider the cross-level spectrum refinement between the two consecutive levels, the $h$-level and the $l$-level. The two operators are $\Theta^h = \mathbf{\Psi}^h\big((\mathbf{\Psi}^h)^T A \mathbf{\Psi}^h\big)^{-1}(\mathbf{\Psi}^h)^T$ and $\Theta^l = \mathbf{\Psi}^l\big((\mathbf{\Psi}^l)^T A \mathbf{\Psi}^l\big)^{-1}(\mathbf{\Psi}^l)^T$ respectively. We have the relations

$$\mathbf{\Psi}^l = \mathbf{\Psi}^h \Psi^l, \quad \mathcal{U}^l = \mathbf{\Psi}^h U^l,$$

$$A^l_{st} = (\mathbf{\Psi}^l)^T A \mathbf{\Psi}^l = (\Psi^l)^T (\mathbf{\Psi}^h)^T A \mathbf{\Psi}^h \Psi^l = (\Psi^l)^T A^h_{st} \Psi^l,$$
$$B^l_{st} = (\mathcal{U}^l)^T A \mathcal{U}^l = (U^l)^T (\mathbf{\Psi}^h)^T A \mathbf{\Psi}^h U^l = (U^l)^T A^h_{st} U^l,$$
$$(A^h_{st})^{-1} = \Psi^l (A^l_{st})^{-1} (\Psi^l)^T + U^l (B^l_{st})^{-1} (U^l)^T,$$

$$\Theta^h = \mathbf{\Psi}^h (\Psi^l (A^l_{st})^{-1} (\Psi^l)^T + U^l (B^l_{st})^{-1} (U^l)^T) (\mathbf{\Psi}^h)^T = \Theta^l + \mathcal{U}^l (B^l_{st})^{-1} (\mathcal{U}^l)^T. \tag{4.5}$$

Now suppose that we have obtained the first $m_l$ essential eigenpairs $(\mu_{l,i}, v_{l,i})$, $i = 1, \cdots, m_l$, of $\Theta^l$. We want to use these eigenpairs as initial guess to obtain the first $m_h$ essential eigenpairs of $\Theta^h$. Recall that we have the estimates

$$|\mu_{h,i} - \mu_{l,i}| \leq \varepsilon_l, \quad 1 \leq i \leq m_l,$$

and

$$\|\Theta^h v_{l,i} - \mu_{h,i} v_{l,i}\|_2 \leq 2\varepsilon_l, \quad 1 \leq i \leq m_l,$$

where $\varepsilon_l$ is the compression error bound. These estimates give us confidence that we can obtain $(\mu_{h,i}, v_{h,i})$, $i = 1, \cdots, m_h$, efficiently from $(\mu_{l,i}, v_{l,i})$, $i = 1, \cdots, m_l$, by using some refinement technique.

Indeed, we will use the Orthogonal Iteration with Ritz Acceleration as our refinement method. Consider an initial guess $Q^{(0)}$ of the first $m$ eigenvectors of a PD operator $\Theta$. To obtain more accurate eigenvalues and eigenspace, the Orthogonal Iteration with Ritz Acceleration runs as follows:

$Q^{(0)} \in \mathbb{R}^{n \times m}$ given with $(Q^{(0)})^T Q^{(0)} = I_m$
$F^{(0)} = \Theta Q^{(0)}$
**for** $k = 1, 2, \cdots$
  $Q^{(k)} R^{(k)} = F^{(k-1)}$ (QR factorization)
  $F^{(k)} = \Theta Q^{(k)}$ $(*)$
  $S^{(k)} = (Q^{(k)})^T F^{(k)}$
  $P^{(k)} D^{(k)} (P^{(k)})^T = S^{(k)}$ (Schur decomposition)
  $Q^{(k)} \leftarrow Q^{(k)} P^{(k)}$
  $F^{(k)} \leftarrow F^{(k)} P^{(k)}$
**end**

To state the convergence property of the Orthogonal Iteration with Ritz Acceleration, we first define the distance between two spaces. Let $V_1, V_2 \subset \mathbb{R}^n$ be two linear spaces, and $\boldsymbol{P}_{V_1}, \boldsymbol{P}_{V_2}$ be the orthogonal projections onto $V_1, V_2$ respectively. We define the distance between $V_1$ and $V_2$ as

$$\text{dist}(V_1, V_2) = \|\boldsymbol{P}_{V_1} - \boldsymbol{P}_{V_2}\|_2.$$

We also use the same notation $\text{dist}(V_1, V_2)$ when $V_1, V_2$ are matrices of column vectors. In this case $\text{dist}(V_1, V_2)$ means $\text{dist}(\text{span}\{V_1\}, \text{span}\{V_2\})$.

Suppose that the diagonal entries $\mu_i^{(k)}$, $i = 1, \cdots, m$, of $D^{(k)}$ are in a decreasing order, then $\mu_i^{(k)}$ is a good approximation of the $i^{th}$ eigenvalue of $\Theta$, and $\text{span}\{Q_i^{(k)}\}$ is a good approximation of the eigenspace spanned by the first $i$ eigenvectors of $\Theta$, where $Q_i^{(k)}$ denotes the first $i$ columns of $Q^{(k)}$. We would like to emphasize that the meaning of the superscript $(k)$ of $\mu_i^{(k)}$ is different from those in Section 4.3. More precisely, we have the following convergence estimate:

**Theorem (Stewart [117], 1968):** Let $(\mu_i, v_i)$, $i = 1, \cdots, N$, be the ordered (essential) eigenpairs of $\Theta$, and let $\mu_i^{(k)}$, $i = 1, \cdots, m$, be the ordered eigenvalues of $D^{(k)} = (Q^{(k)})^T \Theta Q^{(k)}$ given in the Orthogonal Iteration with Ritz Acceleration $(*)$. Let $V_m = [v_1, v_2, \cdots, v_m]$, and $d^{(0)} = \text{dist}(V_m, Q^{(0)})$. Then we have

$$|\mu_i - \mu_i^{(k)}| \leq O\Big((\frac{\mu_{m+1}}{\mu_i})^{2k} \cdot \|\Theta\|_2 \cdot \frac{(d^{(0)})^2}{1 - (d^{(0)})^2}\Big), \quad 1 \leq i \leq m.$$

Moreover, we have

$$\text{dist}(V_m, Q^{(k)}) \leq O\Big((\frac{\mu_{m+1}}{\mu_m})^k \cdot \frac{d^{(0)}}{\sqrt{1 - (d^{(0)})^2}}\Big),$$

and for $i = 1, \cdots, m-1$, if we further assume that $\alpha_i = \mu_i - \mu_{i+1} > 0$, then we have

$$\begin{aligned}\text{dist}(V_i, Q_i^{(k)}) \leq\ & O\Big((\frac{\mu_{m+1}}{\mu_i})^k \cdot \frac{d^{(0)}}{\sqrt{1 - (d^{(0)})^2}}\Big) \\ & + O\Big(\frac{\sqrt{i}}{\alpha_i} \cdot (\frac{\mu_{m+1}^2}{\mu_m \mu_i})^k \cdot \|\Theta\|_2 \cdot \frac{(d^{(0)})^2}{1 - (d^{(0)})^2}\Big),\end{aligned}$$

where $V_i$ and $Q_i^{(k)}$ are the first $i$ columns of $V_m$ and $Q^{(k)}$ respectively.

Now we go back to our problem, where we have $\Theta = \Theta^h$, $m = m_l$, and $Q^{(0)} = V_{m_l}^l = [v_{l,1}, \cdots, v_{l,m_l}]$. We next consider the efficiency of this refinement technique

in our problem. As long as the initial distance $d^{(0)} = \text{dist}(V_{m_l}^h, V_{m_l}^l) < 1$, the first $m_h$ eigenvalues and the eigenspace of the first $m_h$ eigenvectors of $\Theta^h$ converges exponentially fast at a rate $(\frac{\mu_{h,m_l+1}}{\mu_{h,m_h}})^k$. We can expect that a few iterations of refinement will be sufficient to give an accurate eigenspace for narrowing down the residual spectrum of $\Theta^h$, if we can ensure that the ratio $\frac{\mu_{h,m_l+1}}{\mu_{h,m_h}}$ is small enough. This will be verified in our numerical examples to be presented in Section 4.6. In particular, to refine the first $m_h$ eigenpairs subject to a prescribed accuracy $\epsilon$, we need $K = O(\log(\frac{1}{\epsilon})/\log(\frac{\mu_{h,m_h}}{\mu_{h,m_l+1}}))$ refinement iterations.

The main cost of the refinement procedure comes from the computation of $\Theta^h Q^{(0)}$ and the computation of $\Theta^h Q^{(k)}$ in each iteration. We will reduce the computational cost by using the fact that $Q^{(k)}$ is a good approximation of eigenvectors of $\Theta^h$. We first consider how to compute $\Theta^h Q^{(0)}$ efficiently.

Notice that in our problem, we take $Q^{(0)} = V_{m_l}^l$, whose columns are the first $m_l$ eigenvectors of $\Theta^l$. Therefore by (4.5), we have

$$\Theta^h Q^{(0)} = \Theta^h V_{m_l}^l = \Theta^l V_{m_l}^l + \mathcal{U}^l (B_{st}^l)^{-1} (\mathcal{U}^l)^T V_{m_l}^l = V_{m_l}^l D_{m_l}^l + \mathcal{U}^l (B_{st}^l)^{-1} (\mathcal{U}^l)^T V_{m_l}^l,$$

where $D_{m_l}^l$ is a diagonal matrix whose diagonal entries are $\mu_{l,1}, \mu_{l,2}, \cdots, \mu_{l,m_l}$. Recall that by Lemma 3.1.1 and Corollary 3.1.4, $\kappa(B_{st}^l)$ is bounded by $\varepsilon_l \delta_h$ that can be well controlled in the decomposition procedure. Thus it is efficient to solve $(B_{st}^l)^{-1}$ using the CG method. As we have mentioned before, applying $(U^l)^T$ or $U^l$ from the left is performed by doing patch-wise Householder transformations that involve only one local Householder vector on each patch, which takes $O(N^h)$ computational cost, where $N^h$ is the compressed dimension on level $h$ or the size of $A_{st}^h$. Therefore in the CG method, the cost of matrix multiplication of $B_{st}^l = (U^l)^T A_{st}^h U^l$ mainly comes from the number of nonzero entries of $A_{st}^h$. Then the total computational cost of computing $\Theta^h Q^{(0)}$ subject to a relative error $\epsilon$ can be bounded by

$$O\left(m_l \cdot nnz(A_{st}^h) \cdot \varepsilon_l \delta_h \cdot \log(\frac{1}{\epsilon})\right).$$

Next, we consider how to compute $\Theta^h Q^{(k)}$. To do so, we first compute $w_i^{(k)} = (\mathbf{\Psi}^h)^T q_i^{(k)}$, where $q_i^{(k)}$ is the $i^{th}$ column of $Q^{(k)}$, then compute $(A_{st}^h)^{-1} w_i^{(k)}$, and apply $\mathbf{\Psi}^h$. Again we will use the PCG method with predictioner $M^h = (\mathbf{\Psi}^h)^T \mathbf{\Psi}^h$ to compute $(A_{st}^h)^{-1} w_i^{(k)}$. As we have discussed in Section 4.3, this is equivalent to using the CG method to compute $(A_{\mathbf{\Psi}}^h)^{-1} z_i^{(k)}$, where $A_{\mathbf{\Psi}}^h = (M^h)^{-\frac{1}{2}} A_{st}^h (M^h)^{-\frac{1}{2}}$, and $z_i^{(k)} = (M^h)^{-\frac{1}{2}} w_i^{(k)} = (M^h)^{-\frac{1}{2}} (\mathbf{\Psi}^h)^T q_i^{(k)}$. Inspired by Corollary 4.3.6, we seek to provide

a good initial guess for the CG method to ensure efficiency. In the Orthogonal Iteration with Ritz Acceleration (∗), one can check that $(Q^{(k)})^T(\Theta^h Q^{(k)} - Q^{(k)}D^{(k)}) = \mathbf{0}$, where $D^{(k)}$ is a diagonal matrix with diagonal entries $\mu_1^{(k)}, \mu_2^{(k)}, \cdots, \mu_{m_l}^{(k)}$, and therefore

$$\begin{aligned}
&(Z^{(k)})^T((A_{\mathbf{\Psi}}^h)^{-1}Z^{(k)} - Z^{(k)}D^{(k)}) \\
&= (Q^{(k)})^T\mathbf{\Psi}^h(M^h)^{-\frac{1}{2}}\left((A_{\mathbf{\Psi}}^h)^{-1}(M^h)^{-\frac{1}{2}}(\mathbf{\Psi}^h)^T Q^{(k)} - (M^h)^{-\frac{1}{2}}(\mathbf{\Psi}^h)^T Q^{(k)}D^{(k)}\right) \\
&= (Q^{(k)})^T\left(\mathbf{\Psi}^h(A_{st}^h)^{-1}(\mathbf{\Psi}^h)^T Q^{(k)} - \mathbf{\Psi}^h(M^h)^{-1}(\mathbf{\Psi}^h)^T Q^{(k)}D^{(k)}\right) \\
&= (Q^{(k)})^T\left(\Theta^h Q^{(k)} - Q^{(k)}D^{(k)}\right) \\
&= \mathbf{0},
\end{aligned}$$

where we have used that $Q^{(k)} \in \text{span}\{\mathbf{\Psi}^h\}$ and so $\mathbf{\Psi}^h(M^h)^{-1}(\mathbf{\Psi}^h)^T Q^{(k)} = Q^{(k)}$. This observation implies that if we use $\mu_i^{(k)} z_i^{(k)}$ as the initial guess for computing $(A_{\mathbf{\Psi}}^h)^{-1} z_i^{(k)}$ using the CG method, the initial residual $z_i^{(k)} - (A_{\mathbf{\Psi}}^h)(\mu_i^{(k)} z_i^{(k)})$ is orthogonal to $(A_{\mathbf{\Psi}}^h)^{-1}Z^{(k)}$. Since $Q^{(k)}$ are already good approximate essential eigenvectors of $\Theta^h$, $Z^{(k)}$ are good approximate eigenvectors of $(A_{\mathbf{\Psi}}^h)^{-1}$, we can expect that the target eigenspace $Z_{m_h}$, namely the eigenspace of the first $m_h$ eigenvectors of $(A_{\mathbf{\Psi}}^h)^{-1}$, can be well spanned in $\text{span}\{(A_{\mathbf{\Psi}}^h)^{-1}Z^{(k)}\}$. Therefore we can reasonably assume that $z_i^{(k)} - (A_{\mathbf{\Psi}}^h)(\mu_i^{(k)} z_i^{(k)}) \in Z_{m_h^+} = Z_{m_h}^{\perp}$, and so again we can benefit from the restricted condition number $\kappa(A_{\mathbf{\Psi}}^h, Z_{m_h^+}) \leq \mu_{h,m_h+1}\delta_h$ as introduced in Section 4.3. Moreover, we notice that the spectral residual $\|\Theta^h q_i^{(k)} - \mu_i^{(k)} q_i^{(k)}\|_2$ is bounded by $2\varepsilon_l$ by Lemma 4.2.3, and we have

$$\|(A_{st}^h)^{-1}w_i^{(k)} - \mu_i^{(k)}(M^h)^{-1}w_i^{(k)}\|_2 \leq \|(A_{\mathbf{\Psi}}^h)^{-1}z_i^{(k)} - \mu_i^{(k)}z_i^{(k)}\|_2 = \|\Theta^h q_i^{(k)} - \mu_i^{(k)}q_i^{(k)}\|_2, \tag{4.6}$$

where we have used $\lambda_{min}(M^h) \geq 1$ (Lemma 4.3.7). Thus if we use $\mu_i^{(k)} z_i^{(k)}$ as the initial guess, the initial error will be bounded by $2\varepsilon_l$ at most, and the CG procedure will only need

$$O\left(\kappa(A_{\mathbf{\Psi}}^h, Z_{m_h^+}) \cdot \log(\frac{\varepsilon_l}{\epsilon})\right) = O\left(\mu_{h,m_h+1}\delta_h \cdot \log(\frac{\varepsilon_l}{\epsilon})\right)$$

iterations to achieve a relative accuracy $\epsilon$, instead of $O(\kappa(A_{\mathbf{\Psi}}^h, Z_{m_h^+}) \cdot \log(\frac{1}{\epsilon}))$. Notice that using the initial guess $\mu_i^{(k)} z_i^{(k)}$ for $(A_{\mathbf{\Psi}}^h)^{-1} z_i^{(k)}$ is equivalent to using the initial guess $\mu_i^{(k)}(M^h)^{-1}w_i^{(k)}$ for $(A_{st}^h)^{-1}w_i^{(k)}$.

Supported by the analysis above, we will compute $(A_{st}^h)^{-1}w_i^{(k)}$ using the preconditioned CG method with preconditioner $M^h$ and initial guess $\mu_i^{(k)}(M^h)^{-1}w_i^{(k)}$. Again suppose that in each PCG iteration, we also use the CG method to apply $(M^h)^{-1}$

subject to a higher relative accuracy $\hat{\epsilon}$, which takes $O(nnz(M^h) \cdot \kappa(M^h) \cdot \log(\frac{1}{\hat{\epsilon}}))$ computational cost. In practice, it is sufficient to take $\hat{\epsilon}$ comparable to $\epsilon$. Recall that $nnz(M^h) \leq nnz(A^h_{st})$, and $\kappa(M^h) \leq O(\varepsilon_h \delta_h)$ (Lemma 4.3.7), the cost of computing $\Theta^h Q^{(k)}$ subject to a relative error $\epsilon$ is then bounded by

$$O\left(m_l \cdot \mu_{h,m_h+1}\delta_h \cdot \log(\frac{\varepsilon_l}{\epsilon}) \cdot nnz(A^h_{st}) \cdot \varepsilon_h \delta_h \cdot \log(\frac{1}{\epsilon})\right).$$

Notice that in each refinement iteration we also need to perform one QR factorization and one Schur decomposition, which together cost $O(N^h \cdot m_l^2)$. However, as we have mentioned in the introduction, we only consider the asymptotic complexity of our method when the original $A$ becomes super large. In this case, the number $m_{tar}$ of the target eigenpairs is considered as a fixed constant, and so the term $O(N^h \cdot m_l^2) \leq O(N^h m_{tar}^2)$ is considered to be minor and will be omitted in our complexity analysis. Therefore, the total cost of refining the first $m_h$ eigenpairs subject to a prescribed accuracy $\epsilon$ can be bounded by

$$\begin{aligned} &O\left(m_l \cdot nnz(A^h_{st}) \cdot \varepsilon_l \delta_h \cdot \log(\frac{1}{\epsilon})\right) \\ &+ O\left(m_l \cdot \mu_{h,m_h+1}\delta_h \cdot \log(\frac{\varepsilon_l}{\epsilon}) \cdot nnz(A^h_{st}) \cdot \varepsilon_h \delta_h \cdot \log(\frac{1}{\epsilon}) \cdot \log(\frac{1}{\epsilon}) / \log(\frac{\mu_{h,m_h}}{\mu_{h,m_l+1}})\right). \end{aligned} \tag{4.7}$$

Again we remark that the operator $\Theta^h$, the long vectors $Q^{(k)}$, $F^{(k)}$, $V^l$ and $V^h$ are only for analysis use. Operations on long vectors of size $n$ will be very expensive and unnecessary, especially on lower levels where the compression dimension $N^h$(the size of $A^h_{st}$) is small. Notice that all long vectors on the $h$-level are in span$\{\mathbf{\Psi}^h\}$ as

$$Q^{(k)} = \mathbf{\Psi}^h \widehat{Q}^{(k)}, \quad F^{(k)} = \mathbf{\Psi}^h \widehat{F}^{(k)}, \quad V^l_{m_l} = \mathbf{\Psi}^h \widehat{V}^l_{m_l}, \quad V^h_{m_h} = \mathbf{\Psi}^h \widehat{V}^h_{m_h},$$

we thus only operate on their coefficients in the basis $\mathbf{\Psi}^h$. Correspondingly, whenever we need to consider orthogonality of long vectors, we replace it by the $M^h$-orthogonality of their coefficient vectors. One can check that all discussions above still apply. Also another advantage of using the coefficient vectors is that in the previous discussions, the good initial guess $\mu_i^{(k)}(M^h)^{-1}w_i^{(k)} = \mu_i^{(k)}(M^h)^{-1}(\mathbf{\Psi}^h)^T q_i^{(k)} = \mu_i^{(k)}\hat{q}^{(k)}$ is obtained explicitly.

Summarizing the analysis above, we propose the following Algorithm 12 as our refinement method. Since we want the eigenspace spanned by the first $m_h$ eigenvectors of $\Theta^h$ to be computed accurately, the refinement stops when dist$(Q^{(k-1)}_{m_h}, Q^{(k)}_{m_h}) < \epsilon$

for some prescribed accuracy $\epsilon$, where $Q^{(k)}_{m_h}$ denotes the first $m_h$ columns of $Q^{(k)}$. Since $Q^{(k)}$ is orthogonal, one can check that

$$\begin{aligned}
\text{dist}(Q^{(k-1)}_{m_h}, Q^{(k)}_{m_h}) &= \|Q^{(k)}_{m_h} - Q^{(k-1)}_{m_h}(Q^{(k-1)}_{m_h})^T Q^{(k)}_{m_h}\|_2 \\
&= \|\widehat{Q}^{(k)}_{m_h} - \widehat{Q}^{(k-1)}_{m_h}(\widehat{Q}^{(k-1)}_{m_h})^T M^h \widehat{Q}^{(k)}_{m_h}\|_{M^h} \\
&\le \sqrt{\lambda_{max}(M^h)}\|\widehat{Q}^{(k)}_{m_h} - \widehat{Q}^{(k-1)}_{m_h}(\widehat{Q}^{(k-1)}_{m_h})^T M^h \widehat{Q}^{(k)}_{m_h}\|_2 \\
&\le \sqrt{1+\varepsilon_h\delta_h}\|\widehat{Q}^{(k)}_{m_h} - \widehat{Q}^{(k-1)}_{m_h}(\widehat{Q}^{(k-1)}_{m_h})^T M^h \widehat{Q}^{(k)}_{m_h}\|_F.
\end{aligned}$$

In practical, we use $\|\widehat{Q}^{(k)}_{m_h} - \widehat{Q}^{(k-1)}_{m_h}(\widehat{Q}^{(k-1)}_{m_h})^T M^h \widehat{Q}^{(k)}_{m_h}\|_F < \frac{\epsilon}{\sqrt{1+\varepsilon_h\delta_h}}$ as the stopping criterion since it is easy to check. We have used Lemma 4.3.7 to bound $\lambda_{max}(M^h)$.

**Algorithm 12** Eigenpair Refinement

**Input:** $\widehat{V}^l_{m_l}, D^l_{m_l}$, prescribed accuracy $\epsilon$, target eigenvalue threshold $\mu_h$.
**Output:** $\widehat{V}^h_{m_h}, D^h_{m_h}$.

Set $\widehat{Q}^{(0)} = V^l_{m_l}, \quad D^{(0)} = D^l_{m_l}, \quad k = 0;$
**for** $i = 1 : m_l$ **do**
  $g_i = pcg(B^l_{st}, (U^l)^T M^h \hat{q}^{(0)}_i, -, -, \epsilon); \quad (\widehat{Q} = [\hat{q}_1, \cdots, \hat{q}_{m_l}])$
**end for**
$\widehat{F}^{(0)} = \widehat{Q}^{(0)} D^{(0)} + U^l G; \quad (G = [g_1, \cdots, g_{m_l}])$
**repeat**
  $k \leftarrow k + 1;$
  $\widehat{Q}^{(k)} R^{(k)} = \widehat{F}^{(k-1)};$ (QR factorization with respect to $M^h$ orthogonality, i.e. $(\widehat{Q}^{(k)})^T M^h \widehat{Q}^{(k)} = I$)
  $W^{(k)} = M^h \widehat{Q}^{(k)};$
  **for** $i = 1 : m_l$ **do**
    $\hat{f}^{(k)}_i = pcg(A^h_{st}, w^{(k)}_i, M^h, \mu^{(k-1)}_i \hat{q}^{(k)}_i, \epsilon); \quad (\widehat{F} = [\hat{f}_1, \cdots, \hat{f}_{m_l}])$
  **end for**
  $S^{(k)} = (W^{(k)})^T \widehat{F}^{(k)};$
  $P^{(k)} D^{(k)} (P^k)^T = S^{(k)}$ (Schur decomposition, diagonals of $D^{(k)}$ in decreasing order);
  renew $m_h$ so that $\mu^{(k)}_{m_h} \ge \mu_h > \mu^{(k)}_{m_h+1}$;
  $\widehat{Q}^{(k)} \leftarrow \widehat{Q}^{(k)} P^{(k)}, \quad \widehat{F}^{(k)} \leftarrow \widehat{F}^{(k)} P^{(k)};$
**until** $\|\widehat{Q}^{(k)}_{m_h} - \widehat{Q}^{(k-1)}_{m_h}(\widehat{Q}^{(k-1)}_{m_h})^T M^h \widehat{Q}^{(k)}_{m_h}\|_F < \epsilon.$
$\widehat{V}^h_{m_h} = \widehat{Q}^{(k)}_{m_h}, \quad D^h_{m_h} = D^{(k)}_{m_h}.$ ($D^{(k)}_{m_h}$ denotes the first $m_h$-size block of $D^{(k)}$)

### 4.5 Overall Algorithms

Combining the refinement method and the extension method, we now propose our overall Algorithm 13 for computing partial eigenpairs of a PD matrix $A$. It utilizes the a priori multiresolution decomposition of $A$ to compute the first $m_{tar}$ eigenpairs of $A^{-1}$, by passing approximate eigenpairs from lower levels to higher levels to

finally reach a prescribed accuracy. In particular, this algorithm starts with the eigen decomposition of the lowest level (whose dimension is small enough), refines and extends the approximate eigenpairs on each level, and stops at the highest level. The overall accuracy is achieved by the prescribed compression error of the highest level. It could be clearer using a flow chart (Figure 4.1) to illustrate the procedure of our method. If we see the eigenproblem of the original matrix $A$ as a complicated model, our algorithm resolves the model complexity by hierarchically simplifying/coarsening the original model into an inductive sequence of approximate models.

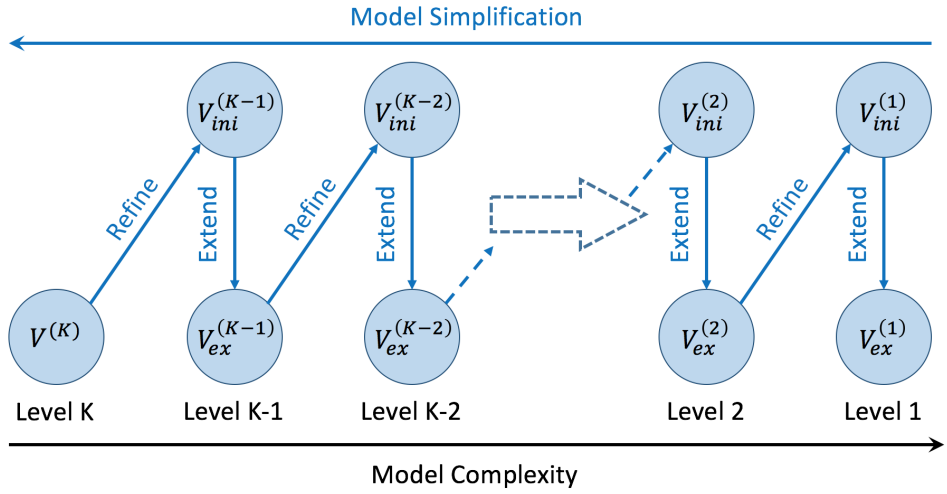



Figure 4.1: A flow chart illustrating the procedure of Algorithm 13.

Recall that the output $\widehat{V}_{ex}^{(k)}$ of the extension process and the initializing process are the coefficients of $V_{ex}^{(k)}$ in the basis $\mathbf{\Psi}^{(k)}$. When passing these results from level $k$ to level $k-1$, we need to recover the coefficients of $V_{ex}^{(k)}$ in the basis $\mathbf{\Psi}^{(k-1)}$. This can be done by simply reforming $\widehat{V}_{ex}^{(k)} \leftarrow \Psi^{(k)}\widehat{V}_{ex}^{(k)}$(Line 3 in Algorithm 13), since $V_{ex}^{(k)} = \mathbf{\Psi}^{(k)}\widehat{V}_{ex}^{(k)} = \mathbf{\Psi}^{(k-1)}\Psi^{(k)}\widehat{V}_{ex}^{(k)}$.

In Algorithm 13, the parameters should be chosen carefully to ensure computational efficiency, by using the analysis in the previous sections. We shall discuss the choice of each parameter separately. To be consistent, we first clarify some notations. Let $\hat{m}_k, m_k$ be the numbers of output eigenpairs of the refinement process and the extension process respectively on level $k$. Ignoring numerical errors, let $(\mu_i^{(k)}, v_i^{(k)})$, $i = 1, \cdots, N^{(k)}$, be the essential eigenpairs of the operator $\Theta^{(k)}$ as in Section 4.3. Let $(\mu_i^{(k)}, v_i^{(k)})$, $i = 1, \cdots, m_k$, denote the output eigenpairs on level $k$. Notice that $(\mu_i^{(k)}, v_i^{(k)})$, $i = 1, \cdots, \hat{m}_k$, are the output of the refinement process, and $(\mu_i^{(k)}, v_i^{(k)})$, $i = \hat{m}_k + 1, \cdots, m_k$, are the output of the extension process.

**Choice Of Multi-level Accuracies $\{\epsilon^{(k)}\}$:** Notice that there is a compression error $\varepsilon_k$ between level $k$ and level $k-1$. That is to say, no matter how accurately we compute the eigenpairs of $\Theta^{(k)}$, they are approximations of eigenpairs of $\Theta^{(k-1)}$ subject to accuracy no better that $\varepsilon_k$. Therefore, on the one hand, the choice of the algorithm accuracy $\epsilon^{(k)}$ for the eigenpairs of $\Theta^{(k)}$ on each level should not compromise the compression error. On the other hand, the accuracy should not be over-achieved due to the presence of the compression error. Therefore, we choose $\epsilon^{(k)} = 0.1 \times \varepsilon_k$ in practice.

**Choice Of Thresholds $\{(\mu_{re}^{(k)}, \mu_{ex}^{(k)})\}_{k=1}^{K}$:** These thresholds provide control on the smallest eigenvalues of output eigenpairs of both the refinement process and the extension process in that

$$\mu_{\hat{m}_k}^{(k)} \geq \mu_{re}^{(k)} > \mu_{\hat{m}_k+1}^{(k)}, \quad \mu_{ex}^{(k)} \geq \mu_{m_k}^{(k)}, \quad k = 1, 2, \cdots, K.$$

Recall that the outputs of the refinement process are the inputs of the extension process, and the outputs of the extension process are the inputs of the refinement process on the higher level. By Theorem 4.3.8, to ensure the efficiency of the extension process, we need to uniformly control the restricted condition number

$$\kappa(A_{\Psi}^{(k)}, Z_{\hat{m}_k^+}^{(k)}) \leq \mu_{\hat{m}_k+1}^{(k)} \delta_k < \mu_{re}^{(k)} \delta_k.$$

Recall that in Section 4.4 the convergence rate of the refinement process is given by $\frac{\mu_{h,m_l+1}}{\mu_{h,m_h}}$, where $l$ corresponds to $k+1$ and $h$ corresponds to $k$ on each level $k$. Thus to ensure the efficiency of the refinement process we need to uniformly control the ratio

$$\frac{\mu_{m_{k+1}+1}^{(k)}}{\mu_{\hat{m}_k}^{(k)}} \leq \frac{\mu_{m_{k+1}}^{(k)}}{\mu_{re}^{(k)}} \leq \frac{\mu_{m_{k+1}}^{(k+1)} + \varepsilon_{k+1}}{\mu_{re}^{(k)}} \leq \frac{\mu_{ex}^{(k+1)} + \varepsilon_{k+1}}{\mu_{re}^{(k)}},$$

where $\varepsilon_{k+1}$ is the compression error between level $k+1$ and level $k$, and we have used Lemma 4.2.3. Thus, more precisely, we need to choose thresholds $\{(\mu_{re}^{(k)}, \mu_{ex}^{(k)})\}_{k=1}^{K}$ so that there exist uniform constants $\kappa > 0, \gamma \in (0,1)$ so that

$$\text{(i) } \mu_{re}^{(k)} \delta_k \leq \kappa, \qquad \text{(ii) } \frac{\mu_{ex}^{(k+1)} + \varepsilon_{k+1}}{\mu_{re}^{(k)}} \leq \gamma. \tag{4.8}$$

Due to the existence of $\varepsilon_k$, condition (ii) implies that there is no need to choose $\mu_{ex}^{(k)}$ much smaller than $\varepsilon_k$, which suffers from over-computing but barely improves the efficiency of the refinement process. So one convenient way is to choose

$$\mu_{re}^{(k)} = \alpha \varepsilon_{k+1}, \qquad \mu_{ex}^{(k)} = \beta \varepsilon_k, \tag{4.9}$$

for some uniform constants $\alpha, \beta > 0$ such that $\alpha > 1 + \beta$. Recall that when constructing the multiresolution decomposition, we impose conditions $\varepsilon_k \delta_k \leq c$ and $\varepsilon_k = \eta \varepsilon_{k+1}$ for some uniform constants $c > 0$ and $\eta \in (0, 1)$. Thus we have

$$\mu_{re}^{(k)} \delta_k = \frac{\alpha}{\eta} \varepsilon_k \delta_k \leq \frac{\alpha c}{\eta} = \kappa, \qquad \frac{\mu_{ex}^{(k+1)} + \varepsilon_{k+1}}{\mu_{re}^{(k)}} = \frac{1+\beta}{\alpha} = \gamma < 1.$$

**Choice of Searching Step $d$:** In the first part of the extension algorithm, we explore the number $m_k$ so that $\mu_{m_k}^{(k)} \leq \mu_{ex}^{(k)}$, and we do this by setting an exploring step size $d$ and examining the last few eigenvalues every $d$ steps of the Lanczos iteration. The step size $d$ should neither be too large to avoid over computing, nor too small to ensure efficiency. In practical, we choose $d = \min\{\lfloor \frac{\dim \mathbf{\Psi}^{(k)}}{10} \rfloor, \lfloor \frac{m_{tar}}{10} \rfloor\}$.

**Complexity:** Now we summarize the complexity of Algorithm 13 for computing the first $m_{tar}$ largest eigenpairs of $A^{-1}$ for a PD matrix $A \in \mathbb{R}^{n \times n}$ subject to an error $\varepsilon$. Suppose we are provided a $K$-level MMD of $A$ with $\varepsilon_k \delta_k \leq c$, $\varepsilon_k = \eta \varepsilon_{k+1}$, and $\varepsilon_1 = \varepsilon$. In what follows, we will uniformly estimate $nnz(A_{st}^{(k)}) \leq nnz(A)$, $\epsilon^{(k)} \geq \epsilon^{(1)} = 0.1\varepsilon_1$ and $m_k \leq m_{tar}$.

We first consider the complexity of all refinement process. Notice that by our choice $\frac{\varepsilon_{k+1}}{\epsilon^{(k)}} = \frac{\varepsilon_{k+1}}{0.1\epsilon^{(k)}} = \frac{1}{0.1\eta}$, the factor $\log(\frac{\varepsilon_l}{\epsilon})$ in (4.7), which is now $\log(\frac{\varepsilon_{k+1}}{\epsilon^{(k)}})$, can be estimated as $O(\log(\frac{1}{\eta}))$. Since we can will make sure $\frac{\mu_{m_{k+1}+1}^{(k)}}{\mu_{\hat{m}_k}^{(k)}} \leq \gamma$ for some constant $\gamma < 1$, the factor $\log(\frac{\mu_{h,m_h}}{\mu_{h,m_l+1}})$ in (4.7), which is now $\log(\frac{\mu_{\hat{m}_k}^{(k)}}{\mu_{m_{k+1}+1}^{(k)}})$, can be seen as a constant. Also using estimates $\mu_{h,m_h}\delta_h \leq \frac{\alpha c}{\eta} = O(\frac{c}{\eta})$, $\varepsilon_l \delta_h \leq \frac{c}{\eta}$, $\varepsilon_h \delta_h \leq c$ and $\log \frac{1}{\epsilon} = O(\log \frac{1}{\varepsilon})$, we modify (4.7) to obtain the complexity of all $K$-level refinement process

$$O\left(m_{tar} \cdot nnz(A) \cdot \frac{c^2}{\eta} \log(\frac{1}{\eta}) \cdot (\log \frac{1}{\varepsilon})^2 \cdot K\right). \tag{4.10}$$

Next we consider the complexity of all extension process. As we have discussed in Section 4.3, the major cost of the extension process comes from the operation of adding a new vector (the adding operation) to the Lanzcos vectors (Line 7 of Algorithm 10 that happens in Line 3 of Algorithm 11). Using estimates $\mu_m \delta(\mathcal{P}) \leq \frac{\alpha c}{\eta} = O(\frac{c}{\eta})$, $\varepsilon(\mathcal{P})\delta(\mathcal{P}) \leq c$, $\log \frac{1}{\epsilon} = O(\log \frac{1}{\varepsilon})$, we modify (4.4) to obtain the cost of every single call of the adding operation as

$$O\left(\frac{c^2}{\eta} \cdot nnz(A) \cdot (\log \frac{1}{\varepsilon})^2\right).$$

On every level, the indexes contributing to adding operations go from $\hat{m}_k + 1$ to $m_k$. Due to the refinement process, we have $\hat{m}_k \leq m_{k+1}$, and so every single index from 1 to $m_{tar}$ may contribute more than one adding operations. But if we reasonably assume that $\mu_{ex}^{(k+1)} > \mu_{re}^{(k-1)}$, namely $\beta > \alpha\eta$ under parameter choice (4.9), we will have $m^{(k+1)} < \hat{m}^{(k-1)}$, and so every index from 1 to $m_{tar}$ will contribute no more than two adding operations. Therefore the total cost of all extension process can be estimated as

$$O\left(m_{tar} \cdot \frac{c^2}{\eta} \cdot nnz(A) \cdot (\log \frac{1}{\varepsilon})^2\right). \tag{4.11}$$

We remark that the cost of implicit restarting process is only a constant multiple of (4.11). Combining (4.10) and (4.11), we obtain the total complexity of our method

$$O\left(m_{tar} \cdot nnz(A) \cdot \frac{c^2}{\eta} \log(\frac{1}{\eta}) \cdot (\log \frac{1}{\varepsilon})^2 \cdot K\right). \tag{4.12}$$

To further simplify (4.12), we need to use estimates for the MMD given in Section 3.2. In particular, to preserve sparsity $nnz(A_{st}^{(k)}) \leq nnz(A)$, we need to choose the scale ratio $\eta^{-1} = (\log \frac{1}{\varepsilon} + \log n)^p$ as in (3.21) for some constant $p$. We remark that for graph Laplacian, $p = 1$. The resulting level number is $K = O(\frac{\log n}{\log(\log \frac{1}{\varepsilon} + \log n)})$. The condition bound $c$ can be imposed to be uniform constant by Algorithm 6 in Section 3.2.1. Then the overall complexity of Algorithm 13 can be estimated as

$$\begin{aligned} &O\left(m_{tar} \cdot nnz(A) \cdot (\log \frac{1}{\varepsilon} + \log n)^p \cdot (\log \frac{1}{\varepsilon})^2 \cdot \log n\right) \\ &= O\left(m_{tar} \cdot nnz(A) \cdot (\log \frac{1}{\varepsilon} + \log n)^{p+3}\right). \end{aligned} \tag{4.13}$$

## 4.6 Numerical Examples

In this section we present several numerical examples for the eigensolver. We will use Algorithm 13 to compute a relative large number of eigenpairs of large matrices subject to prescribed accuracies.

### 4.6.1 Dataset Description

The datasets we use are drawn from different physical contexts. They are generated as 3D point clouds and transformed into graphs by adding edges in the KNN setting.

- The first dataset is the well-known "Stanford Bunny" from Stanford 3D Scanning Repository[1]. A reconstructed bunny has 35947 vertices that can be embedded into a surface in $\mathbb{R}^3$ with 5 holes in the bottom.

[1] http://graphics.stanford.edu/data/3Dscanrep/

**Algorithm 13** Hierarchical Eigenpair Computation

**Input:** $K$-level decomposition $\{\Theta^{(k)}\}_{k=1}^{K}$ of PD matrix $A$, target number $m_{tar}$, searching step $d$,
prescribed multi-level accuracies $\{\epsilon^{(k)}\}$, extension thresholds $\{\mu_{ex}^{(k)}\}_{k=1}^{K}$, refinement thresholds $\{\mu_{re}^{(k)}\}_{k=1}^{K}$.
**Output:** $V$, $D$.

Find the eigen pairs $[\widehat{V}_{ex}^{(K)}, D_{ex}^{(K)}]$ of the eigen problem $(A_{st}^{(K)})^{-1}M^{(K)}x = \mu x$;
**for** $k = K-1 : 1$ **do**
  $\widehat{V}_{ex}^{(k+1)} \leftarrow \Psi^{(k+1)}\widehat{V}_{ex}^{(k+1)}$
  $[\widehat{V}_{ini}^{(k)}, D_{ini}^{(k)}] = \text{Eigen\_Refine}([\widehat{V}_{ex}^{(k+1)}, D_{ex}^{(k+1)}]; \epsilon^{(k)}, \mu_{re}^{(k)})$;
  $op = OP(\,\cdot\,; A^{(k)}, M^{(k)}, \epsilon^{(k)})$;
  $[\widehat{V}_{ex}^{(k)}, D_{ex}^{(k)}] = \text{Eigen\_Extend}([\widehat{V}_{ini}^{(k)}, D_{ini}^{(k)}]; op, \epsilon^{(k)}, \mu_{ex}^{(k)}, d, m_{tar})$;
**end for**
$V = \Psi^{(1)}\widehat{V}_{ex}^{(1)} \quad D = D_{ex}^{(1)}$.

- The second dataset is a MRI data of brain from the Open Access Series of Imaging Sciences (OASIS)[2]. They use FreeSurfer to reconstruct the surface from MRI scan and obtain a point cloud with 48463 points.

- The third dataset is a "SwissRoll" model, which is popular in manifold learning. Vertices are generated by

$$(x_i, y_i, z_i) = (t_i cos(t_i), y_i, t_i sin(t_i)) + \boldsymbol{\eta}_i,\ i = 1, 2, ..., n, \tag{4.14}$$

  where $t_i \overset{\text{i.i.d}}{\sim} \mathcal{U}[1.5\pi, 4.5\pi]$, $y_i \overset{\text{i.i.d}}{\sim} \mathcal{U}[0, 20]$, and $\eta_i \overset{\text{i.i.d}}{\sim} \mathcal{N}(\mathbf{0}, 0.05I_3)$. It can be viewed as a spiral of one and a half rounds plus random noise. In our examples the roll has $n = 20000$ points.

With point clouds at hand, we apply the KNN setting to construct graphs with $k_{bunny} = 20$, $k_{brain} = 20$ and $k_{swissroll} = 10$. Each existing edge $e_{ij}$ is weighted as $e^{-r_{i,j}^2/\sigma}$, where $r_{i,j}$ is the Euclidean distance between vertices $v_i$ and $v_j$, and $\sigma$ is a parameter. We have $\sigma_{bunny} = 10^{-6}$, $\sigma_{brain} = 10^{-4}$ and $\sigma_{swiss} = 0.1$. Figure 4.2 shows the point clouds of datasets.

From the graphs given above, we construct their related graph Laplacians $L$ in

[2]http://www.oasis-brains.org/

the general setting:

$$L_{ij} = \begin{cases} \sum_{k\sim i} w_{ik}, & i = j, \\ \\ -w_{ij}, & i \neq j. \end{cases}$$

Further, without loss of generality, we rescale all graph Laplacians and add uniform self-loops of weight 1 to them, so that each of them satisfies (i)$\lambda_1 = 1$, (ii) $\lambda_2 = O(1)$. Under this construction, we obtain three graph Laplacian matrices $L_{bunny}, L_{brain}, L_{swissroll}$. $L_{bunny}$ has size $n = 35947$, sparsity $nnz = 714647$ and condition number $\kappa(L_{bunny}) = 1.86 \times 10^4$; $L_{brain}$ has size $n = 48463$, sparsity $nnz = 1038065$ and condition number $\kappa(L_{bunny}) = 1.14 \times 10^5$; $L_{swissroll}$ has size $n = 20000$, sparsity $nnz = 248010$ and condition number $\kappa(L_{bunny}) = 1.15 \times 10^6$.

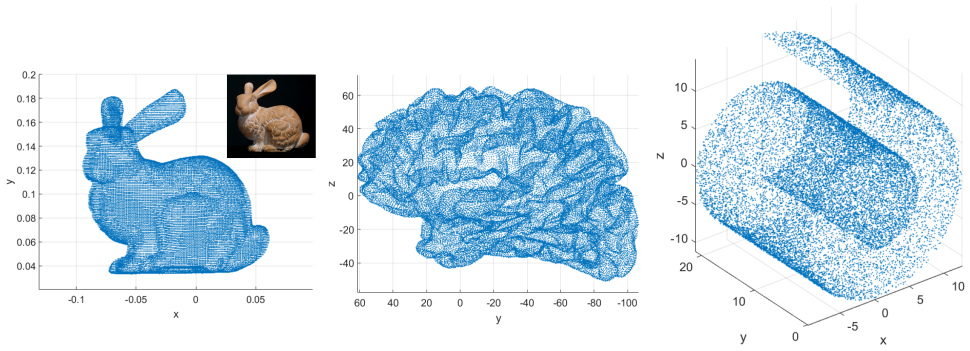


Figure 4.2: Point cloud of datasets. From left to right: (1) bunny (point cloud and sculpture); (2) brain; (3) Swiss roll.

### 4.6.2 Numerical MMD

Before computing eigenpairs of graph Laplacians from our datasets using Algorithm 13, we need to apply Algorithm 6 in Section 3.2.1 to obtain the MMDs, which is the only pre-computation step in our proposed algorithm. For each graph Laplacian, we perform the decomposition with a prescribed condition bound $c$ and a series of multi-level resolutions (compression errors) $\{\varepsilon_k\}_{k=1}^K$. Note that we perform two decompositions with different multiresolutions for the SwissRoll data. The decomposition time for each example is reported in Table 4.1. Recall that the total complexity of Algorithm 6 is $O\left(nnz(A) \cdot \log n \cdot \left(\log \frac{1}{\epsilon} + \log n\right)^{3d+p}\right)$, where $d$ denotes the intrinsic geometric dimension of the graph. By comparing with the complexity estimate in (4.13), when $m_{tar} \gg \left(\log \frac{1}{\epsilon} + \log n\right)^{3d-2}$, the pre-computation time for constructing MMD only takes up a relative small portion of overall time. As illustrated later in Table 4.8, even with the pre-computation time taken into account, our proposed algorithm is still faster than other well-established methods.

| Data | 2-level Bunny | 4-level Brain | 4-level SwissRoll | 3-level SwissRoll |
|---|---|---|---|---|
| Time | 10.245 | 34.589 | 8.124 | 9.430 |

Table 4.1: Matrix decomposition time (in seconds) for different examples.

Table 4.2 and Table 4.3 give the detailed information of all decompositions we will use for eigenpair computation. In Table 4.2, $K$ is the number of levels, $\varepsilon_1$ is the finest (prescribed) accuracy, $\eta$ is the ratio $\varepsilon_k/\varepsilon_{k+1}$ and $c$ is the condition bound such that $\varepsilon_k\delta_k \leq c$. By Lemma 4.3.7, the condition number of $M^{(k)}$ is bounded as $\kappa(M^{(k)}) \leq 1 + \varepsilon_k\delta_k \approx c$, and by Corollary 3.1.4, the condition number of $B^{(k)}$ is bounded as $\kappa(B^{(k)}) \leq \varepsilon_k\delta_{k-1} \leq c/\eta$. We can see in Table 4.3 that these bounds are well satisfied. Recall that the bounded condition number of $M^{(k)}$ is essential for the efficiency of Algorithm 11, and the bounded condition number of $B^{(k)}$ is essential for the efficiency of Algorithm 12.

Table 4.3 also shows the detailed information for all four decompositions. The 2-norm of $A^{(k)}$, namely $\lambda_{max}(A^{(k)})$ decreases as $k$ increases, and well bounded as $\|A^{(k)}\|_2 \leq \delta_k \leq c\varepsilon_k^{-1}$ as expected by Corollary 3.1.4 (we have normalized $\mu_1 = \|L^{-1}\|_2$ to 1). And the sparsities of $A^{(k)}$ and $M^{(k)}$ are of the same order as the sparsity of $A^{(0)} = L$, i.e. $nnz(A^{(k)}), nnz(M^{(k)}) = O(nnz(A^{(0)}))$ as ensured by our choice of parameters discussed in Section 4.5.

| Data | $K$ | $\varepsilon_1$ | $\eta$ | $c$ | Bound on $\kappa(M^{(k)})$ | Bound on $\kappa(B^{(k)})$ |
|---|---|---|---|---|---|---|
| Bunny | 2 | $10^{-3}$ | 0.1 | 20 | 20 | 200 |
| Brain | 4 | $10^{-4}$ | 0.2 | 20 | 20 | 100 |
| SwissRoll | 3 | $10^{-5}$ | 0.1 | 20 | 20 | 200 |
| SwissRoll | 4 | $10^{-5}$ | 0.2 | 20 | 20 | 100 |

Table 4.2: Decomposition parameters different datasets.

### 4.6.3 The Coarse Level Eigenpair Approximation

We first use the decompositions given above to compute the first few eigenpairs of graph Laplacians with relatively low accuracies. Numerical results reveal that even on the coarse levels, the compressed (low dimensional) operators show good spectral approximation properties with regard to the smallest eigenvalues of $L$ (or the largest eigenvalues of $L^{-1}$). Here we take the bunny data and the brain data as examples. For the bunny data, we use the lowest level $k = 2$ with compression error $\varepsilon_2 = 0.01$; for the brain data, we use level $k = 3$ with compression error

| Level $k$ | $\varepsilon_k$ | $\dim(A^{(k)})$ | $nnz(A^{(k)})$ | $\|A^{(k)}\|_2$ | $nnz(M^{(k)})$ | $\kappa(M^{(k)})$ | $\kappa(B^{(k)})$ |
|---|---|---|---|---|---|---|---|
| The 2-level decomposition of Bunny data. | | | | | | | |
| 0 | - | 35947 | $714k = m$ | $1.86 \times 10^4$ | - | - | - |
| 1 | $10^{-3}$ | 2641 | $613k \approx 0.86m$ | $1.05 \times 10^4$ | $203k \approx 0.28m$ | 1.45 | 5.58 |
| 2 | $10^{-2}$ | 198 | $27k \approx 0.04m$ | $1.37 \times 10^3$ | $10k \approx 0.02m$ | 2.05 | 45.03 |
| The 4-level decomposition of Brain data. | | | | | | | |
| 0 | - | 48463 | $1038k = m$ | $1.14 \times 10^5$ | - | - | - |
| 1 | $10^{-4}$ | 11622 | $2546k \approx 2.45m$ | $7.82 \times 10^4$ | $725k \approx 0.70m$ | 1.29 | 5.80 |
| 2 | $5 \times 10^{-4}$ | 1713 | $431k \approx 0.42m$ | $2.01 \times 10^4$ | $189k \approx 0.18m$ | 1.84 | 18.34 |
| 3 | $2.5 \times 10^{-3}$ | 252 | $37k \approx 0.04m$ | $3.33 \times 10^3$ | $20k \approx 0.02m$ | 2.19 | 28.23 |
| 4 | $1.25 \times 10^{-2}$ | 35 | $1k < 0.01m$ | $4.53 \times 10^2$ | $1k < 0.01m$ | 2.02 | 35.08 |
| The 3-level decomposition of SwissRoll data. | | | | | | | |
| 0 | - | 20000 | $248k = m$ | $1.15 \times 10^6$ | - | - | - |
| 1 | $10^{-5}$ | 5528 | $689k \approx 2.78m$ | $4.31 \times 10^5$ | $197k \approx 0.79m$ | 1.45 | 10.06 |
| 2 | $10^{-4}$ | 723 | $108k \approx 0.44m$ | $7.44 \times 10^4$ | $35k \approx 0.14m$ | 2.30 | 67.47 |
| 3 | $10^{-3}$ | 55 | $2k < 0.01m$ | $5.45 \times 10^3$ | $1k < 0.01m$ | 3.92 | 185.93 |
| The 4-level decomposition of SwissRoll data. | | | | | | | |
| 0 | - | 20000 | $248k = m$ | $1.15 \times 10^6$ | - | - | - |
| 1 | $10^{-5}$ | 5528 | $689k \approx 2.78m$ | $4.31 \times 10^5$ | $197k \approx 0.79m$ | 1.45 | 10.06 |
| 2 | $5 \times 10^{-5}$ | 1347 | $215k \approx 0.87m$ | $9.36 \times 10^4$ | $65k \approx 0.26m$ | 1.90 | 26.52 |
| 3 | $2.5 \times 10^{-4}$ | 203 | $18k \approx 0.08m$ | $1.89 \times 10^4$ | $9k \approx 0.04m$ | 3.06 | 98.87 |
| 4 | $1.25 \times 10^{-3}$ | 53 | $2k < 0.01m$ | $3.72 \times 10^3$ | $1k < 0.01m$ | 3.36 | 51.14 |

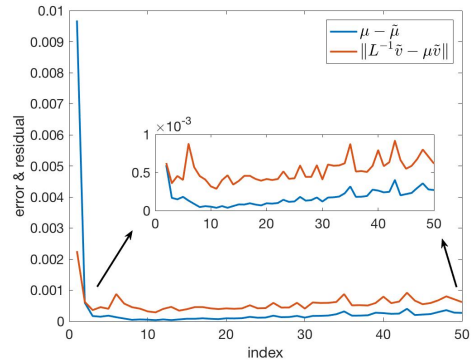
Table 4.3: Decomposition information of (i) Bunny (2-level) (ii) Brain (4-level) and (iii) SwissRoll (3, 4-level) data. $m \triangleq nnz(A^{(0)})$. $1k = 1000$.

$\varepsilon_3 = 0.0025$. We compute the first 50 eigenpairs $\{\tilde{v}_i, \tilde{\lambda}_i\}$ of the compressed operator by directly solving the general eigen problem (Lemma 4.2.2)

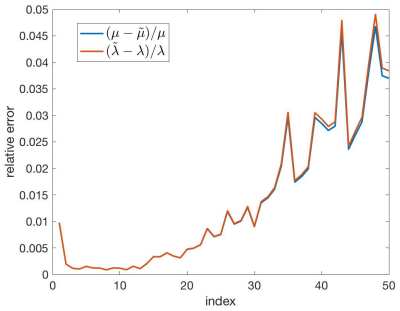
$$A^{(k)} z_i = \tilde{\lambda}_i M^{(k)} z_i, \quad \tilde{v}_i = \mathbf{\Psi}^{(k)} z_i, \quad i = 1, \cdots, 50.$$

The computation of the coarse level eigenproblem is much more efficient due to the compressed dimension. To show the error of the approximate eigenvalues, the ground truth is obtained by using the Eigen C++ Library [3]. Figure 4.3 shows the absolute and relative errors of these eigenvalues. In both cases $\mu_i$ is the $i$th largest eigenvalue of $L^{-1}$ and $\lambda_i = 1/\mu_i$; $\tilde{\mu}_i$ is the $i$th largest eigenvalue of the compressed problem $\Theta^{(k)}$ and $\tilde{\lambda}_i = 1/\tilde{\mu}_i$. By Lemma 4.2.3, $|\mu_i - \tilde{\mu}_i|$ is bounded by $\varepsilon_k$ and $\|L^{-1}\tilde{v}_i - \mu_i \tilde{v}_i\|_2$ is bounded $2\varepsilon_k$. We can see in Figure 4.3 that both estimates are well satisfied. In particular, the error of the first eigenvalue is close to the bound of $\varepsilon_k$. However, the first eigenpair is already known. Therefore, we are only interested in the $2^{\text{nd}}$ up to $50^{\text{th}}$ eigenvalues and we embed the sub-plot of these eigenvalue

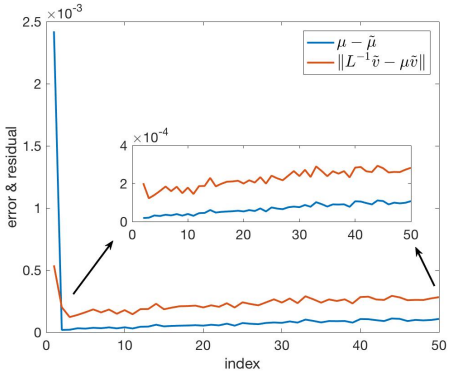
errors as shown in Figure 4.3a and Figure 4.3c respectively.



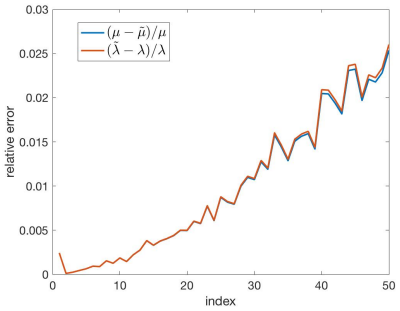

(a) error $\mu_i - \tilde{\mu}_i$ and residual $\|L^{-1}\tilde{v}_i - \mu_i\tilde{v}_i\|_2$




(b) relative error of $\mu_i$ and $\lambda_i$




(c) error $\mu_i - \tilde{\mu}_i$ and residual $\|L^{-1}\tilde{v}_i - \mu_i\tilde{v}_i\|_2$




(d) relative error of $\mu_i$ and $\lambda_i$

Figure 4.3: The error, the residual and the relative error. Top: Bunny data; bottom: Brain data

Next, we qualitatively examine the accuracy of the approximate eigenvectors of the compressed operators by comparing their behaviors in image segmentation to those of the true eigenvectors of the original Laplacian operators. In the image segmentation, the eigenvectors of graph Laplacian provide a solution to graph partitioning problem. Namely, for a partition $(A, B)$ that satisfies $A \cup B = V$ and $A \cap B = \emptyset$, a measure of their disassociation called the normalized cut ($Ncut$) is defined as [107]

$$Ncut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)}, \tag{4.15}$$

where

$$cut(A, B) = \sum_{u \in A, v \in B} w(u, v), \qquad assoc(A, V) = \sum_{u \in A, t \in V} w(u, t).$$

[3]Eigen C++ Library is available at `http://eigen.tuxfamily.org/index.php?title=Main_Page`

Shi and Malik [107] showed that, for a connected graph, minimizing $Ncut$ can be rephrased as finding the eigenvector $v_2$ that corresponds to the second smallest eigenvalue $\lambda_2$ of the graph Laplacian (since we always have $\lambda_1 = 0$ and $v_1$ a uniform vector). Taking $sign(v_2)$ transforms it into a binary vector which gives a satisfactory cut. Moreover, the next few eigenvectors provide further cuts of the previously partitioned fractions. Therefore, our eigensolver may serve as a powerful tool for graph partitioning, as well as its applications including image segmentation and manifold learning.

We test graph partitioning on bunny and brain datasets using the eigenvectors of both original and compressed operators. Figures 4.4 and 4.5 shows the colormap and the partition generated by some selected eigenvectors. From the pictures we can see that the original and the compressed operators give very similar results when it comes to graph partitioning. The compressed operator is not only easier to compute, but also gives a satisfactory partition in practical settings.

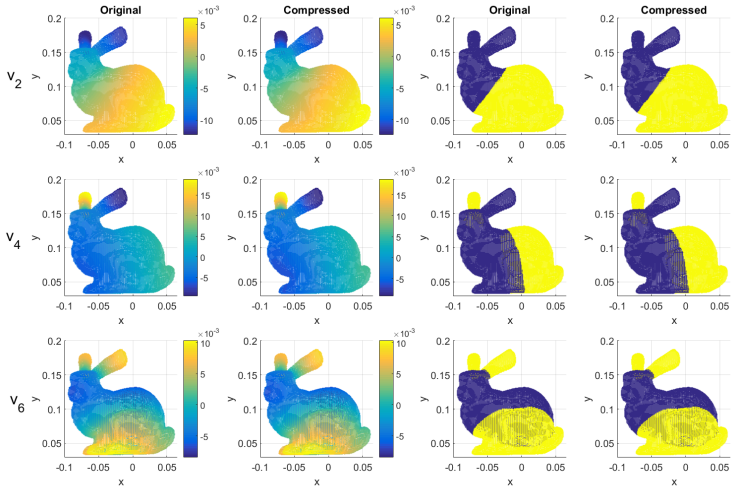



Figure 4.4: Colormap (left) and partition (right) using the 2nd, 4th and 6th eigenvectors of the original/compressed operator.

Figure 4.6 gives an example of refining the partition with more eigenvectors. In the brain data, a fraction that is left intact in the first 5 eigenvectors (the light green part on the left) is divided into a lot more fractions when eigenvectors pile up to 15.

### 4.6.4 The Multi-level Eigenpair Computation

To test the efficacy of the hierarchical structure in our approach, we use our main Algorithm 13 to compute a relatively large number of eigenpairs of Laplacian ma-

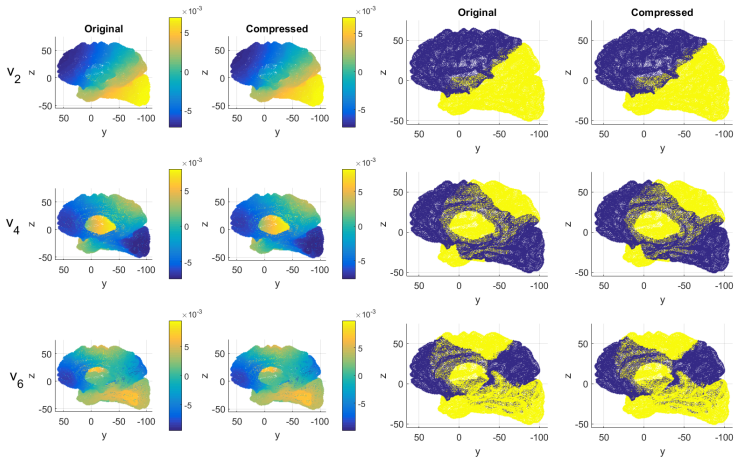



Figure 4.5: Colormap (left) and partition (right) using the 2nd, 4th and 6th eigenvectors of the original/compressed operator.

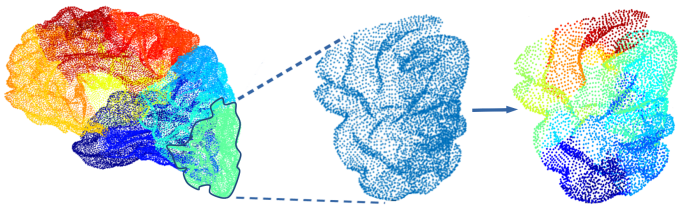


Figure 4.6: Heaping up more eigenvectors leads to finer partition. Left: partition using the first 5 eigenvectors. Middle: a uniform fraction from the previous partition. Right: further partition using the next 10 eigenvectors.

trices subject to the prescribed accuracy. In particular, for both the Brian data and the SwissRoll data, we compute the first 500 eigenpairs of the graph Laplacian subject to prescribed accuracy $|\lambda_i^{-1} - \tilde{\lambda}_i^{-1}| = |\mu_i - \tilde{\mu}_i| \leq \epsilon = \varepsilon_1$. The three decompositions of these two datasets are used in this experiment; for each decomposition, we apply Algorithm 13 with two sets of parameters, $(\alpha, \beta) = (5, 2)$ and $(\alpha, \beta) = (3, 1)$. The details of the results obtained by Algorithm 13 are summarized in Table 4.4-Table 4.7. In Table 4.4, parameters $\alpha, \beta, \kappa, \gamma$ are defined in Section 4.5. In Table 4.5-Table 4.7, we collect numerical results that reflect the efficiency of each single process (refinement or extension). Here we give a detailed description of the notations we use in these tables:

- #I and #O denote the numbers of input and output eigenpairs. To be consistent with the notations defined in Section 4.5, we use (#I,#O)= $(m_{k+1}, \hat{m}_k)$ for refinement process on level $k$, and (#I,#O)= $(\hat{m}_k, m_k)$ for extension process on level $k$.

- #Iter denotes the number of orthogonal iterations in the refinement process. Note that this number is controlled by the ratio $\gamma$.

- $\#_{\text{cg}}(B^{(k)})$ denotes number of CG calls concerning $B^{(k)}$ in the refinement process; $\#_{\text{pcg}}(A^{(k)})$ denotes the number of PCG calls concerning $A^{(k)}$ in the refinement process and the extension process. $\overline{\#}(B^{(k)})$ and $\overline{\#}(A^{(k)})$ denote the average numbers of matrix-vector multiplications concerning $B^{(k)}$ and $A^{(k)}$ respectively, namely the average numbers of iterations, in one single call of CG or PCG. Note that $\overline{\#}(B^{(k)})$ is controlled by $\log(1/\epsilon^{(k)})\kappa(B^{(k)}) \leq \log(1/\epsilon^{(k)})c/\eta$, and $\overline{\#}(A^{(k)})$ by $\log(1/\epsilon^{(k)})\kappa(A^{(k)}_{\mathbf{\Psi}}, Z^{(k)}_{\hat{m}^+_k}) \leq \log(1/\epsilon^{(k)})\alpha c/\eta$.

- As the extension process proceeds, the target spectrum to be computed on this level shrinks even more, and so does the restricted condition number of the operator. Thus the numbers of iterations in each PCG call get much smaller than its expected control $\log(1/\epsilon^{(k)})\alpha c/\eta$, which is a good thing in practice. So to study how the theoretical bound $\log(1/\epsilon^{(k)})\alpha c/\eta$ really affects the efficiency of PCG calls, it is more reasonable to investigate the maximal number of iterations in one PCG call on each level. We use $\widehat{\#}(A^{(k)})$ to denote the largest number of iterations in one single PCG call on level $k$.

- $\overline{\#}(M^{(k)})$ denotes the average number of matrix-vector multiplications concerning $M^{(k)}$ in one single CG call. Such CG calls occur in the PCG calls concerning $A^{(k)}$ where $M^{(k)}$ acts as the preconditioner. Note that $\overline{\#}(M^{(k)})$ is controlled by $\log(1/\epsilon^{(k)})\kappa(M^{(k)}) \leq \log(1/\epsilon^{(k)})(1+c)$.

- "Main Cost" denotes the main computational cost contributed by matrix-vector multiplication flops. In the refinement process we have

$$\begin{aligned}\text{Main Cost} =& \#_{\text{cg}}(B^{(k+1)}) \cdot \overline{\#}(B^{(k+1)}) \cdot nnz(A^{(k)}) \\ &+ \#_{\text{pcg}}(A^{(k)}) \cdot \overline{\#}(A^{(k)}) \cdot \left(nnz(A^{(k)}) + \overline{\#}(M^{(k)}) \cdot nnz(M^{(k)})\right),\end{aligned}$$

  while in the extension process we have

$$\text{Main Cost} = \#_{\text{pcg}}(A^{(k)}) \cdot \overline{\#}(A^{(k)}) \cdot \left(nnz(A^{(k)}) + \overline{\#}(M^{(k)}) \cdot nnz(M^{(k)})\right).$$

Table 4.5-Table 4.7 show the efficiency of our algorithm. We can see that $\overline{\#}(B^{(k)})$ and $\overline{\#}(M^{(k)})$ are well bounded as expected, due to the artificial imposition of the condition bound $c$. $\widehat{\#}(A^{(k)})$ and the numerical condition number $\widehat{\#}(A^{(k)})/\log(1/\epsilon^{(k)})$ are also well controlled by choosing $\alpha$ properly to bound $\kappa = \alpha c/\eta$. It is worth mentioning that $\widehat{\#}(A^{(k)})/\log(1/\epsilon^{(k)})$ appears to be uniformly bounded for all levels, actually

| Data | MMD | $(\alpha, \beta)$ | $(\eta, c)$ | $\kappa$ | $\gamma$ | Total #Iter | Total Main Cost |
|---|---|---|---|---|---|---|---|
| Brain | 4-level | $(5, 2)$ | $(0.2, 20)$ | 500 | 3/5 | 12 | $4.37 \times 10^5 \cdot m$ |
| | 4-level | $(3, 1)$ | $(0.2, 20)$ | 300 | 2/3 | 15 | $4.13 \times 10^5 \cdot m$ |
| SwissRoll | 3-level | $(5, 2)$ | $(0.1, 20)$ | 1000 | 3/5 | 13 | $7.56 \times 10^5 \cdot m$ |
| | 3-level | $(3, 1)$ | $(0.1, 20)$ | 600 | 2/3 | 16 | $7.17 \times 10^5 \cdot m$ |
| SwissRoll | 4-level | $(5, 2)$ | $(0.2, 20)$ | 500 | 3/5 | 19 | $7.00 \times 10^5 \cdot m$ |
| | 4-level | $(3, 1)$ | $(0.2, 20)$ | 300 | 2/3 | 28 | $5.86 \times 10^5 \cdot m$ |

Table 4.4: Eigenpair computation information. $m \triangleq nnz(A^{(0)})$.

much smaller than $\kappa$, which reflects our uniform control on efficiency. #Iter is well bounded due to the proper choice of $\beta$ for bounding $\gamma = (1 + \beta)/\alpha$.

We may also compare the results for the same decomposition but from two different sets of parameters $(\alpha, \beta)$. For all three decompositions, the experiments with $(\alpha, \beta) = (5, 2)$ have a smaller $\gamma = \frac{3}{5}$, and thus is more efficient in the refinement process (less #Iter and less refinement main cost). While the experiments with $(\alpha, \beta) = (3, 1)$ have a smaller $\kappa$ that leads to better efficiency in the extension process (smaller $\widehat{\#}(A^{(k)})/\log(1/\epsilon^{(k)})$ and less extension main cost). But since the dominant cost of the whole process comes from the extension process, thus the experiments with $(\alpha, \beta) = (3, 1)$ have a smaller total main cost.

We remark that the choice of $(\alpha, \beta)$ not only determines $(\kappa, \gamma)$ that will affect the algorithm efficiency, but also determines the segmentation of the target spectrum and its allocation towards different levels of the decomposition. Smaller values of $\alpha$ and $\beta$ means more eigenpairs being computed on coarser levels (larger $k$), which relieves the burden of the extension process for finer levels, but also increases the load of the refinement process. There could be an optimal choice of $(\alpha, \beta)$ that minimizes the total main cost, balancing between the refinement and the extension processes. However, without a priori knowledge of the distribution of the eigenvalues, which is the case in practice, a safe choice of $(\alpha, \beta)$ would be $\alpha, \beta = O(1)$.

To further investigate the behavior of our algorithm, we focus on numerical experiments carried out on the 4-level decomposition of the SwissRoll data. Figure 4.7 shows the convergence of the computed spectrum in different errors. Figure 4.8 shows the completion and the convergence process of the target spectrum in the case of $(\alpha, \beta) = (3, 1)$ (corresponding to Table 4.7). We use a log-scale plot to illustrate the error $|\mu_i - \tilde{\mu}^{(k)}|$ after we complete the refinement process and the

| $(\alpha, \beta) = (5, 2)$ | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Refinement | Lvl $k$ | (#I,#O) | #Iter | $\#_{cg}(B)$ | $\overline{\#}(B)$ | $\#_{pcg}(A)$ | $\overline{\#}(A)$ | $\overline{\#}(M)$ | Main Cost |
| | 3 | (7, 4) | 4 | 7 | 24.43 | 28 | 10.97 | 6.10 | $5.66 \times 10^1 \cdot m$ |
| | 2 | (41, 17) | 4 | 41 | 25.90 | 164 | 16.26 | 6.12 | $4.50 \times 10^3 \cdot m$ |
| | 1 | (207, 84) | 4 | 207 | 23.44 | 828 | 19.17 | 4.64 | $1.02 \times 10^5 \cdot m$ |
| Extension | Lvl $k$ | (#I,#O) | $\widehat{\#}(A)$ | $\epsilon$ | $\frac{\widehat{\#}(A)}{log(1/\epsilon)}$ | $\#_{pcg}(A)$ | $\overline{\#}(A)$ | $\overline{\#}(M)$ | Main Cost |
| | 3 | (4, 41) | 43 | $2.5 \times 10^{-4}$ | 5.18 | 175 | 16.93 | 5.39 | $4.37 \times 10^2 \cdot m$ |
| | 2 | (17, 207) | 75 | $5.0 \times 10^{-5}$ | 7.57 | 500 | 32.27 | 5.47 | $2.27 \times 10^4 \cdot m$ |
| | 1 | (84, 500) | 82 | $10^{-5}$ | 7.12 | 1248 | 44.23 | 4.45 | $3.07 \times 10^5 \cdot m$ |
| $(\alpha, \beta) = (3, 1)$ | | | | | | | | | |
| Refinement | Lvl $k$ | (#I,#O) | #Iter | $\#_{cg}(B)$ | $\overline{\#}(B)$ | $\#_{pcg}(A)$ | $\overline{\#}(A)$ | $\overline{\#}(M)$ | Main Cost |
| | 3 | (15, 6) | 5 | 15 | 24.54 | 75 | 7.74 | 6.07 | $1.08 \times 10^2 \cdot m$ |
| | 2 | (78, 28) | 5 | 78 | 25.85 | 390 | 11.17 | 6.01 | $7.39 \times 10^3 \cdot m$ |
| | 1 | (276, 140) | 5 | 276 | 23.43 | 1380 | 14.28 | 4.67 | $1.29 \times 10^5 \cdot m$ |
| Extension | Lvl $k$ | (#I,#O) | $\widehat{\#}(A)$ | $\epsilon$ | $\frac{\widehat{\#}(A)}{log(1/\epsilon)}$ | $\#_{pcg}(A)$ | $\overline{\#}(A)$ | $\overline{\#}(M)$ | Main Cost |
| | 3 | (6, 78) | 37 | $2.5 \times 10^{-4}$ | 4.46 | 225 | 14.12 | 5.41 | $4.70 \times 10^2 \cdot m$ |
| | 2 | (28, 276) | 57 | $5.0 \times 10^{-5}$ | 5.75 | 600 | 27.91 | 5.43 | $2.34 \times 10^4 \cdot m$ |
| | 1 | (140, 500) | 63 | $10^{-5}$ | 5.47 | 1080 | 42.09 | 4.46 | $2.53 \times 10^5 \cdot m$ |

Table 4.5: 4-level eigenpairs computation of Brain data with $(\eta, c) = (0.2, 20)$.

extension process respectively on each level $k$. As we can see, each application of the refinement process improves the accuracy of the first $\hat{m}_k$ eigenvalues at least by a factor of $\eta = \frac{\varepsilon_k}{\varepsilon_{k+1}}$, but at the price of discarding the last $m_{k+1} - \hat{m}_k$ computed eigenvalues. So the computation of the last $m_{k+1} - \hat{m}_k$ computed eigenvalues on the coarser level $k + 1$ actually serves as preconditioning to ensure the efficiency of the refinement process on level $k$. Then the extension process extends the spectrum to $m_k$ that is determined by the threshold $\mu_{ex}^{(k)}$. The whole computation is an iterative process that improves the accuracy of the eigenvalues by applying the hierarchical Lanczos method to each eigenvalue at most twice.

We also further verify our critical control of the restricted condition number $\kappa(A_{\mathbf{\Psi}}^{(k)}, Z_{\hat{m}_k^+}^{(k)})$ by the parameter $\kappa = \alpha c/\eta$, by showing the dependence of $\widehat{\#}(A^{(k)})$ (or $\widehat{\#}(A^{(k)})/\log(1/\epsilon^{(k)})$) on $\kappa$. Recall that $\widehat{\#}(A^{(k)})$ denotes the largest number of iterations in one single PCG call concerning $A^{(k)}$ on level k. Using the 4-level decomposition of the SwissRoll data with $(\eta, c) = (0.2, 20)$, we perform Algorithm 13 with

| $(\alpha, \beta) = (5, 2)$ | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Refinement | Lvl $k$ | (#I,#O) | #Iter | $\#_{cg}(B)$ | $\overline{\#}(B)$ | $\#_{pcg}(A)$ | $\overline{\#}(A)$ | $\overline{\#}(M)$ | Main Cost |
| | 2 | (21, 12) | 7 | 21 | 52.14 | 147 | 17.61 | 6.33 | $3.91 \times 10^3 \cdot m$ |
| | 1 | (232, 100) | 6 | 232 | 47.23 | 1392 | 16.08 | 5.29 | $1.86 \times 10^5 \cdot m$ |
| Extension | Lvl $k$ | (#I,#O) | $\widehat{\#}(A)$ | $\epsilon$ | $\frac{\widehat{\#}(A)}{log(1/\epsilon)}$ | $\#_{pcg}(A)$ | $\overline{\#}(A)$ | $\overline{\#}(M)$ | Main Cost |
| | 2 | (12, 232) | 94 | $10^{-5}$ | 8.16 | 650 | 28.20 | 7.25 | $2.67 \times 10^4 \cdot m$ |
| | 1 | (100, 500) | 101 | $10^{-6}$ | 7.31 | 1200 | 59.44 | 6.10 | $5.42 \times 10^5 \cdot m$ |
| $(\alpha, \beta) = (3, 1)$ | | | | | | | | | |
| Refinement | Lvl $k$ | (#I,#O) | #Iter | $\#_{cg}(B)$ | $\overline{\#}(B)$ | $\#_{pcg}(A)$ | $\overline{\#}(A)$ | $\overline{\#}(M)$ | Main Cost |
| | 2 | (35, 19) | 8 | 35 | 51.89 | 280 | 13.13 | 6.45 | $5.74 \times 10^3 \cdot m$ |
| | 1 | (315, 165) | 8 | 315 | 46.85 | 2520 | 12.73 | 5.37 | $2.66 \times 10^5 \cdot m$ |
| Extension | Lvl $k$ | (#I,#O) | $\widehat{\#}(A)$ | $\epsilon$ | $\frac{\widehat{\#}(A)}{log(1/\epsilon)}$ | $\#_{pcg}(A)$ | $\overline{\#}(A)$ | $\overline{\#}(M)$ | Main Cost |
| | 2 | (19, 315) | 69 | $10^{-5}$ | 5.99 | 700 | 25.10 | 7.29 | $2.57 \times 10^4 \cdot m$ |
| | 1 | (165, 500) | 78 | $10^{-6}$ | 5.65 | 1005 | 54.91 | 6.11 | $4.20 \times 10^5 \cdot m$ |

Table 4.6: 3-level eigenpairs computation of SwissRoll data with $(\eta, c) = (0.1, 20)$.

fixed $\beta = 1$ but different $\alpha \in [3, 5]$. Figure 4.9 shows $\widehat{\#}(A^{(k)})$ versus $\alpha$ for all three levels. By Theorem 4.3.8, we expect that $\widehat{\#}(A^{(k)}) \propto \kappa \cdot \log(1/\epsilon^{(k)}) \propto \alpha \cdot \log(1/\epsilon^{(k)})$. This linear dependence is confirmed in Figure 4.9. It is also important to note that the curve(green) corresponding to level 1 is below the curve (blue) corresponding to level 2 in Figure 4.9b, which again implies that $\widehat{\#}(A^{(k)})/\log(1/\epsilon^{(k)})$ is uniformly bounded for all levels.

## 4.7 Comparison with the IRLM

Owning to the observation in [76] that IRLM is still one of the most performing and well-known algorithms for finding a large portion of smallest eigenpairs, in this section, we compare the computation complexity of our proposed algorithm with the IRLM.

To quantitatively compare the two methods, we record the computation time and the number of CG iterations as the benchmarks. The reasons for doing this are as follows:

- In large-scale setting, direct method for solving sparse matrix $A^{-1}$ is general, not practical since large memory storage is required. Instead, iterative methods, especially the CG method (as $A$ is PD in our case) is employed.

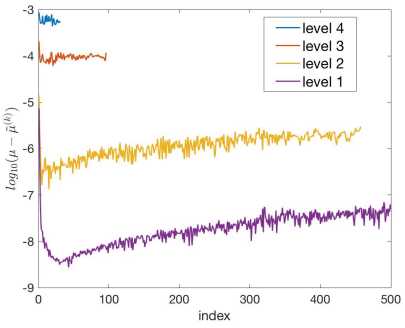
| $(\alpha, \beta) = (5, 2)$ | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Refinement | Lvl $k$ | (#I,#O) | #Iter | $\#_{cg}(B)$ | $\overline{\#}(B)$ | $\#_{pcg}(A)$ | $\overline{\#}(A)$ | $\overline{\#}(M)$ | Main Cost |
| | 3 | (18, 10) | 6 | 18 | 22.61 | 108 | 7.19 | 7.87 | $3.39 \times 10^2 \cdot m$ |
| | 2 | (84, 44) | 8 | 84 | 43.45 | 672 | 10.42 | 6.49 | $2.11 \times 10^4 \cdot m$ |
| | 1 | (390, 195) | 5 | 390 | 28.85 | 1950 | 11.68 | 5.42 | $1.92 \times 10^5 \cdot m$ |
| Extension | Lvl $k$ | (#I,#O) | $\widehat{\#}(A)$ | $\epsilon$ | $\frac{\widehat{\#}(A)}{log(1/\epsilon)}$ | $\#_{pcg}(A)$ | $\overline{\#}(A)$ | $\overline{\#}(M)$ | Main Cost |
| | 3 | (10, 84) | 42 | $2.5 \times 10^{-5}$ | 3.96 | 200 | 18.32 | 8.43 | $1.53 \times 10^3 \cdot m$ |
| | 2 | (44, 390) | 63 | $5 \times 10^{-6}$ | 5.16 | 1050 | 29.30 | 7.24 | $8.47 \times 10^4 \cdot m$ |
| | 1 | (195, 50) | 71 | $10^{-6}$ | 5.13 | 915 | 57.47 | 6.10 | $4.00 \times 10^5 \cdot m$ |
| $(\alpha, \beta) = (3, 1)$ | | | | | | | | | |
| Refinement | Lvl $k$ | (#I,#O) | #Iter | $\#_{cg}(B)$ | $\overline{\#}(B)$ | $\#_{pcg}(A)$ | $\overline{\#}(A)$ | $\overline{\#}(M)$ | Main Cost |
| | 3 | (31, 16) | 7 | 31 | 22.45 | 217 | 6.09 | 8.09 | $5.89 \times 10^2 \cdot m$ |
| | 2 | (95, 67) | 12 | 95 | 43.44 | 1140 | 7.66 | 6.66 | $2.63 \times 10^4 \cdot m$ |
| | 1 | (459, 314) | 7 | 459 | 28.75 | 3656 | 8.71 | 5.56 | $2.65 \times 10^5 \cdot m$ |
| Extension | Lvl $k$ | (#I,#O) | $\widehat{\#}(A)$ | $\epsilon$ | $\frac{\widehat{\#}(A)}{log(1/\epsilon)}$ | $\#_{pcg}(A)$ | $\overline{\#}(A)$ | $\overline{\#}(M)$ | Main Cost |
| | 3 | (16, 95) | 31 | $2.5 \times 10^{-5}$ | 2.92 | 200 | 16.61 | 8.48 | $1.39 \times 10^3 \cdot m$ |
| | 2 | (67, 459) | 49 | $5 \times 10^{-6}$ | 4.01 | 1100 | 25.66 | 7.27 | $7.79 \times 10^4 \cdot m$ |
| | 1 | (314, 500) | 55 | $10^{-6}$ | 3.98 | 558 | 50.61 | 6.12 | $2.15 \times 10^5 \cdot m$ |

Table 4.7: 4-level eigenpairs computation of SwissRoll data: $(\eta, c) = (0.2, 20)$.

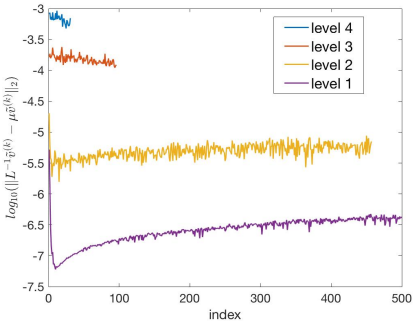
- In both the IRLM and our proposed algorithm, the dominating complexity comes from the operator of computing $A^{-1}b$ for some $b$.

**Remark 4.7.1.** *For small-scale problems, a direct solver (such as sparse Cholesky factorization) for $A^{-1}$ is preferred in the IRLM. In this way, only one factorization step for A is required prior to the IRLM. Moreover, solving for $A^{-1}$ in each iteration is replaced by solving two lower triangular matrix systems. This will bring a significant speedup for the IRLM. However, recall that we are aiming at understanding the asymptotic behavior and performance of these methods. Therefore, the IRLM discussed in this section employs the iterative solver instead of a direct solver.*

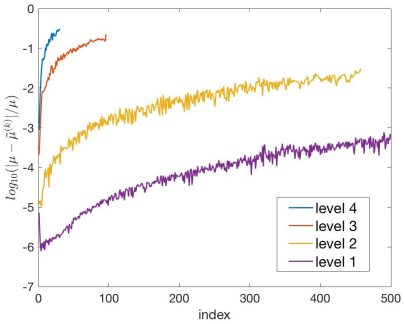
To be consistent, all the experiments are performed on a single machine equipped with Intel(R) Core(TM) i5-4460 CPU with 3.2GHz and 8GB DDR3 1600MHz RAM. Both the proposed algorithm and the IRLM are implemented using C++ with



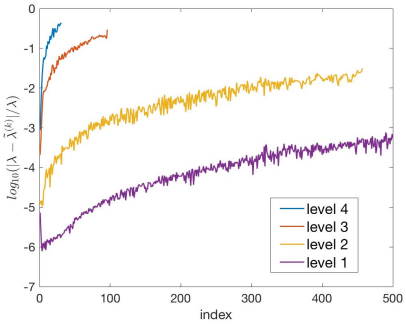

(a) $\log_{10}(\mu_i - \tilde{\mu}_i^{(k)})$, $i = 1, \cdots, m_k$




(b) $\log_{10}(\|L^{-1}\tilde{v}_i^{(k)} - \mu_i \tilde{v}_i^{(k)}\|_2)$, $i = 1, \cdots, m_k$




(c) $\log_{10}((\mu_i - \tilde{\mu}_i^{(k)})/\mu_i)$, $i = 1, \cdots, m_k$



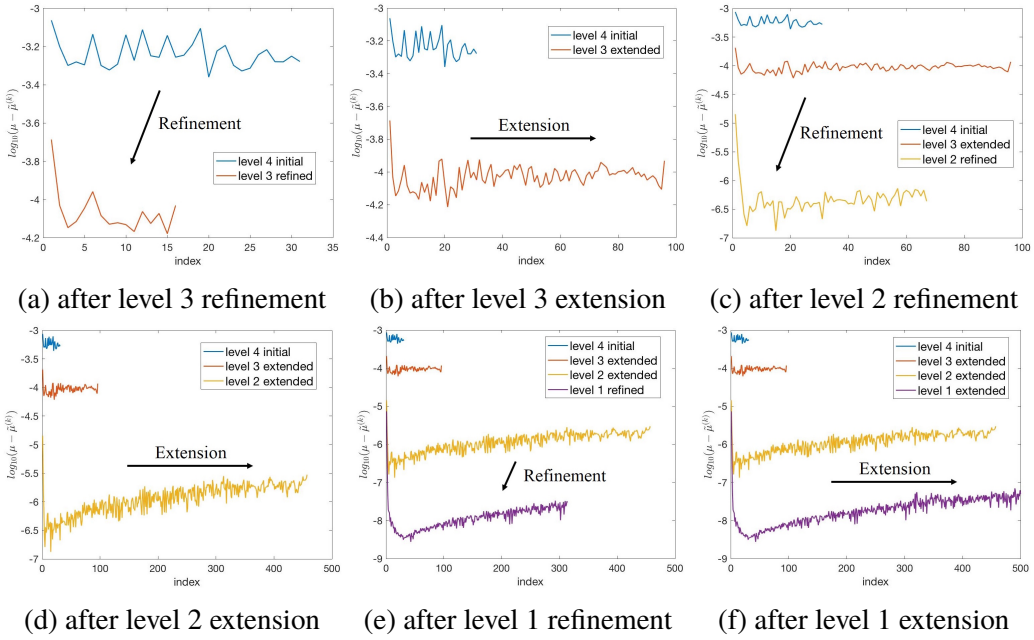

(d) $\log_{10}((\lambda_i - \tilde{\lambda}_i^{(k)})/\lambda_i)$, $i = 1, \cdots, m_k$

Figure 4.7: Convergence of computed spectrum in different errors.

the Eigen Library for fairness. In particular, the built-in (preconditioned) CG solvers are used in the IRLM implementation, instead of implementing on our own.

Table 4.8 shows the overall computation time for computing the leftmost (i) 300; (ii) 200 and (iii) 100 eigenpairs using (i) our proposed algorithm, (ii) the IRLM with incomplete Cholesky PCG method (IRLM-ICCG); and (iii) the IRLM with classical CG method (IRLM-CG). In this numerical example, the error tolerance of the eigenvalues in all three cases is set to $10^{-5}$. Since the error for IRLM cannot be obtained a priori, we fine-tune the relative error tolerance for the (preconditioned) CG solver such that eigenvalues errors are of order $O(10^{-6})$. For the proposed algorithm, the time required for level-wise eigenpair computation is recorded. In the bottom level (level-4 or level-3 in these cases), we have used the built-in eigensolver function in the Eigen Library to obtain the full eigenpairs (corresponding to Line 1 in Algorithm 6). As the problem size is small, the time complexity is insignificant for all three examples.

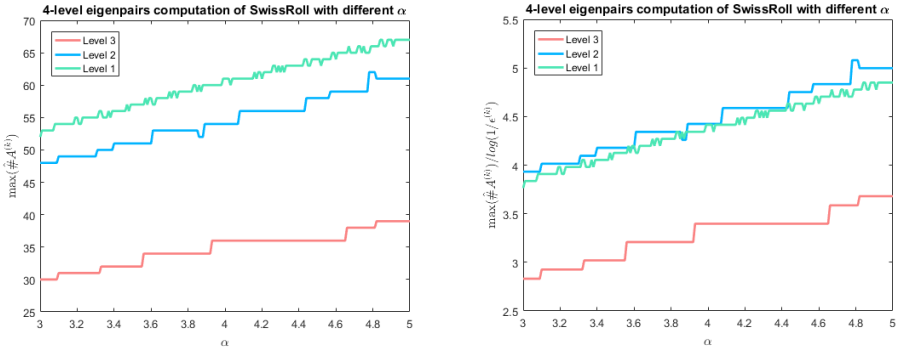
The total runtime of our proposed algorithm in each example is computed by sum-




(a) after level 3 refinement (b) after level 3 extension (c) after level 2 refinement

(d) after level 2 extension (e) after level 1 refinement (f) after level 1 extension

Figure 4.8: The completion and convergence process of the target spectrum. The refinement process retains part of the spectrum subject to threshold $\mu_{re}^{(k)}$ with improved accuracy, and the extension process extends the spectrum subject to threshold $\mu_{ex}^{(k)}$. The whole process is an iterative procedure that aims at improving the accuracy of the eigenvalue solver.




(a) $\hat{\#}A^{(k)}$ versus $\alpha$ (b) $\hat{\#}A^{(k)}/\log(\epsilon_{pcg}^{-1})$ versus $\alpha$

Figure 4.9: $\hat{\#}A^{(k)}$ versus $\alpha$ in the 4-level SwissRoll example.

ming up all levels' computation time, plus the MMD time (which is the second row in Table 4.8). For all these examples, our proposed algorithm outperforms the IRLM. Although both the size of the matrices and their corresponding condition numbers are not extremely large, the numerical experiments already show a observable improvement. From the theoretical analysis discussed in the previous sections, this improvement will even be magnified if the PD matrices are of larger scales and

| # Eigenpairs | Methods | | 4-level Brain | 4-level SwissRoll | 3-level SwissRoll |
|---|---|---|---|---|---|
| | Decomposition* | | 34.589 | 8.124 | 9.430 |
| 300 | Proposed** | Level-4 | 0.010 | 0.011 | - |
| | | Level-3 | 0.841 | 0.560 | 0.083 |
| | | Level-2 | 29.122 | 40.796 | 18.729 |
| | | Level-1 | 61.286 | 18.846 | 22.440 |
| | Total ( $*+**$ ) | | 125.848 | 68.337 | 50.682 |
| | IRLM-CG | | 174.028 | 81.005 | |
| | IRLM-ICCG | | 525.73 | 289.385 | |
| 200 | Proposed** | Level-4 | 0.010 | 0.011 | - |
| | | Level-3 | 0.826 | 0.526 | 0.083 |
| | | Level-2 | 25.560 | 28.094 | 11.517 |
| | | Level-1 | 54.951 | 12.107 | 18.378 |
| | Total ( $*+**$ ) | | 115.936 | 48.862 | 39.408 |
| | IRLM-CG | | 124.871 | 61.479 | |
| | IRLM-ICCG | | 417.632 | 196.217 | |
| 100 | Proposed** | Level-4 | 0.010 | 0.011 | - |
| | | Level-3 | 0.831 | 0.531 | 0.083 |
| | | Level-2 | 25.056 | 22.062 | 9.883 |
| | | Level-1 | 31.882 | 8.066 | 12.029 |
| | Total ( $*+**$ ) | | 92.368 | 38.794 | 31.425 |
| | IRLM-CG | | 115.676 | 48.713 | |
| | IRLM-ICCG | | 324.648 | 90.175 | |

Table 4.8: Computation time (in seconds) for the 4-level Brain, 3-level SwissRoll and the 4-level SwissRoll examples using the proposed Hierarchical multi-level eigensolver; the IRLM with CG solver and the IRLM with incomplete Cholesky PCG solver.

more ill conditioned. Indeed, we assert that our proposed algorithm cannot be fully utilized in these illustrations. Therefore, one of the main future works is to perform detailed numerical experiments in these cases. For instance, by considering the 3-level and 4-level SwissRoll examples, we observe that a 3-level decomposition is indeed sufficient for SwissRoll graph Laplacian, where we recall the corresponding condition number is $\|A\|_2 = 1.15 \times 10^6$. Therefore, using a 3-level decomposition, the overall runtime reduction goes up to approximately 37% when 300 eigenpairs are required.

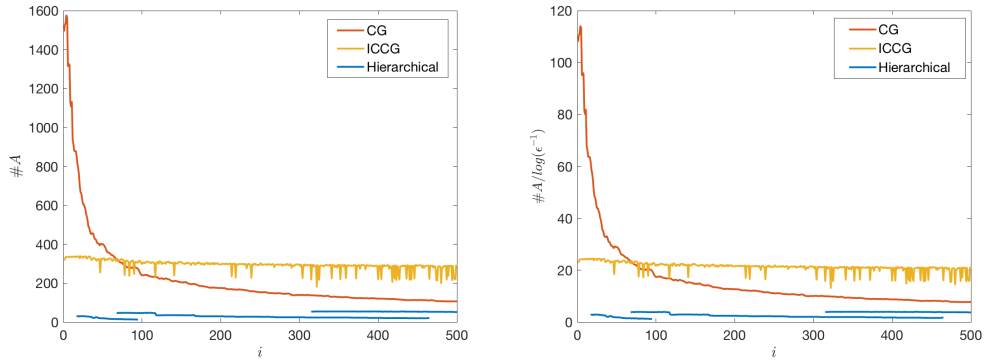
Notice that the time required for the IRLM-ICCG is notably much more than that of the IRLM-CG, which contradicts to our usual experience regarding preconditioning. In fact, such phenomenon can be explained as follows: In the early stage of the IRLM, preconditioning with incomplete Cholesky factorization helps reducing the

iteration number of the CG. However, when the eigen-subspace is gradually projected away throughout the IRLM process, the spectrum of the remaining subspace reduces and therefore CG iteration numbers also drops significantly. On the contrary, preconditioning with incomplete Cholesky ignores such update in spectrum and therefore the CG iteration number is uniform throughout the whole Lanczos iteration. Hence, the classical CG method is preferred if a large number of leftmost eigenpairs are required. Figure 4.10a shows the CG iteration numbers in the IRLM-ICCG, IRLM-CG and respectively, our proposed hierarchical eigensolver versus the Lanczos iteration. More precisely, if we call $V_k$ in (4.1) to be the *Lanczos vector*, the x-axis in the figure then corresponds to the first time we generate the $i$-th column of the Lanczos vector. For IRLM methods, it is equivalent to the extension procedure for the $i$-th column of the Lanczos vector, which corresponds to Line 6 – 8 in Algorithm 8. In particular, the CG iteration number recorded in this figure corresponds to the operation *op* in Line 8 of Algorithm 8. For our proposed algorithm, there are three separate sections, each section's CG iteration numbers correspond to the formation of Lanczos vectors in the $3^{\text{rd}}$-, $2^{\text{nd}}$- and the $1^{\text{st}}$-level respectively. Since we may also update some of these Lanczos vector during the refinement process, therefore some overlaps in the recording of CG iteration numbers corresponding to those Lanczos vector are observed. With the spectrum-preserving hierarchical preconditioner $M$ introduced in our algorithm, the CG iteration number for computing $A^{-1}b$ for some $b$ is tremendously reduced. In contrast, the CG iteration number for IRLM-CG is the largest at the beginning but decreases exponentially and asymptotically converges to our proposed result. For IRLM-ICCG, the incomplete Cholesky factorization does not capture the spectrum update and therefore the iteration numbers is uniform throughout the computation. This observation is also consistent to the time complexity as shown in Table 4.8. Figure 4.10b shows the corresponding normalized plot, where the iteration number is normalized by $\log(\frac{1}{\epsilon})$.

Similar results can also be plotted for the 4-level Brain and the 3-level SwissRoll examples. We therefore skip those plots to avoid repetition.

## 4.8 On Compressed Eigenproblems

In this section, we compare the our method for compressed eigenproblem and the method proposed by Ozoliņš et al. [90]. We start with the straightforward compression directly using the eigenvectors corresponding to smallest eigenvalues,




(a) (b)

Figure 4.10: (a) The PCG iteration number in the 4-level SwissRoll example. The IRLM-ICCG methods exhibits a uniform iteration number, while the IRLM-ID has an exponential decaying iteration number. For our proposed algorithm, since the spectrum-preserving hierarchical preconditioner $M$ is employed, the CG iteration number is minimum. This is also consistent to the time complexity shown in Table 4.8. (b) The corresponding normalized plot, where the iteration number is normalized by $\log(\epsilon)$.

which can be obtained by solving the following optimization problem:

$$\begin{aligned} \Psi \;=\; & \arg\min_{\widehat{\Psi}} \; \textstyle\sum_{i=1}^{N} \hat{\psi}_i^T A \hat{\psi}_i, \\ & \text{s.t.} \quad \hat{\psi}_i^T \hat{\psi}_j = \delta_{ij}, \; i, j = 1, 2, \cdots, N. \end{aligned} \tag{4.16}$$

The compression via eigenvectors is well known as the PCA method is optimal in 2-norm sense for fixed compressed dimension $N$. However, computing a large number of eigenvectors is a hard problem itself, not to mention that we actually intend to approximate eigenpairs using the compressed operator. Also the spatially extended profiles of exact eigenvectors make them less favorable in many fields of researches. Then as modification, Ozoliņš et al. [90] added a $L_1$ regularization term to impose the desired locality on $\Psi$. They modified the optimization problem (4.16) as

$$\begin{aligned} \Psi \;=\; & \arg\min_{\widehat{\Psi}} \; \textstyle\sum_{i=1}^{N} \left( \hat{\psi}_i^T A \hat{\psi}_i + \frac{1}{\mu} \|\hat{\psi}_i\|_1 \right), \\ & \text{s.t.} \quad \hat{\psi}_i^T \hat{\psi}_j = \delta_{ij}, \; i, j = 1, 2, \cdots, N. \end{aligned} \tag{4.17}$$

The $L_1$ regularization, as widely used in many optimization problems for sparsity pursuit, effectively ensures each output $\psi_i$ to have spatially compact support, at the cost of compromising the approximation accuracy compared to PCA. The factor $\mu$ controls the locality of $\Psi$. A smaller $\mu$ gives more localized profiles of $\Psi$, which, however, results in larger compression error for a fixed $N$. The loss of approximation

accuracy can be compensated by increasing, yet not significantly, the basis number $N$. An algorithm based on the split Bregman iteration was also proposed in [90] to effectively solve the problem (4.17). In summary, their work provides an effective method to find a bunch of localized basis functions that can approximately span the eigenspace of smallest eigenvalues of $A$.

Although our approach to operator compression is originally developed from a different perspective based on FEM, it can be reformulated as an optimization problem similar to (4.16). In fact, to obtain the basis $\Psi$ used in our method, we can simply replace the nonlinear constraints $\psi_i^T \psi_j = \delta_{ij},\ i,j = 1,2,\cdots,N$, by linear constraints $\psi_i^T \phi_j = \delta_{ij},\ i,j = 1,2,\cdots,N$, to get

$$\begin{aligned} \Psi \ = \ & \arg\min_{\widehat{\Psi}} \ \textstyle\sum_{i=1}^{N} \hat{\psi}_i^T A \hat{\psi}_i, \\ & \text{s.t.} \quad \hat{\psi}_i^T \phi_j = \delta_{ij},\ i,j = 1,2,\cdots,N. \end{aligned} \tag{4.18}$$

Here $\Phi = [\phi_1, \phi_2, \cdots, \phi_N]$ is a dual basis that we construct ahead of $\Psi$ to provide a priori compression error estimate as stated in Theorem 2.2.1. As the constraints become linear, problem (4.18) can be solved explicitly by $\Psi = A^{-1}\Phi(\Phi^T A^{-1}\Phi)^{-1}$ as mentioned in (2.24). Instead of imposing locality by adding $L_1$ regularization as in (4.17), we obtain the exponential decaying feature of $\Psi$ by constructing each dual basis function $\phi_i$ locally. That is the locality of $\Phi$ and the strong correlation $\Psi^T\Phi = I$ automatically give us the locality of $\Psi$ under energy minimizing property. The optimization form (4.18) was derived by Owhadi in [88] where $\Psi$ was used as the FEM basis to solve second-order elliptic equations with rough coefficients. This methodology was then generalized to problems on higher-order elliptic equations [53], general Banach space [89] and general sparse PD matrix [52]. In all previous works the nice spectral property of $\Psi$ has been observed and in particular the eigenspace corresponding to the smallest $M$ eigenvalues of $A$ can be approximately well spanned by $\Psi$ of a relative larger dimension $N = O(M)$.

To further compare the problems (4.17) and (4.18), we test both of them on the one-dimensional Kronig–Penney (KP) model studied in [90] with rectangular potential wells replaced by inverted Gaussian potentials. In this example, the matrix $A$ comes from discretization of the PDE operator $-\frac{1}{2}\Delta + V(x)$ defined on the domain $\Omega$ with periodic boundary condition. In particular, $\Omega = [0, 50]$, and $V(x) = -V_0 \sum_{j=1}^{N_{el}} \exp\big(-\frac{(x-x_j)^2}{2\delta^2}\big)$. As in [90], we discretize $\Omega$ with 512 equally spaced nodes, and we choose $N_{el} = 5$, $V_0 = 1$, $\delta = 3$, and $x_j = 10j - 5$(instead of $x_j = 10j$ in [90], which essentially changes nothing).

For problem (4.18), we divide $\Omega$ into $N$ equal-length intervals $\{\Omega_i\}_{i=1}^N$, and choose the dual basis $\Phi = [\phi_1, \phi_2, \cdots, \phi_N]$ such that $\phi_i$ is the discretization of the indicator function $\mathbf{1}(\Omega_i)(\mathbf{1}(\Omega_i)(x) = 1$ for $x \in \Omega_i$, otherwise $\mathbf{1}(\Omega_i)(x) = 0)$. We use $\Psi_o$ to denote the exact result of problem (4.18), namely $\Psi_o = A^{-1}\Phi(\Phi^T A^{-1}\Phi)^{-1}$. Since $\Psi_o$ is not orthogonal, we should compute the eigenvalues from the general eigenvalue problem $\Psi_o^T A\Psi_o v = \lambda\Psi_o^T\Psi_o v$(Lemma 4.2.2) as approximations of the eigenvalues of $A$. We use $\lambda_o$ to denote these approximate eigenvalues.

For problem (4.17), we use Algorithm 1 and exactly the same parameters provided in [90], which means we are simply reproducing their results, except that we use a finer discretization (512 rather than 128) and we shift the potential $V(x)$. We have used normalized $\Phi$ as the initial guess for Algorithm 1 in [90], and choose $\mu = 10$. We use $\Psi_{cm}$ to denote the result of problem (4.17). We use $\lambda_{cm}$ to denote the eigenvalues of $\Psi_{cm}^T A\Psi_{cm}$.

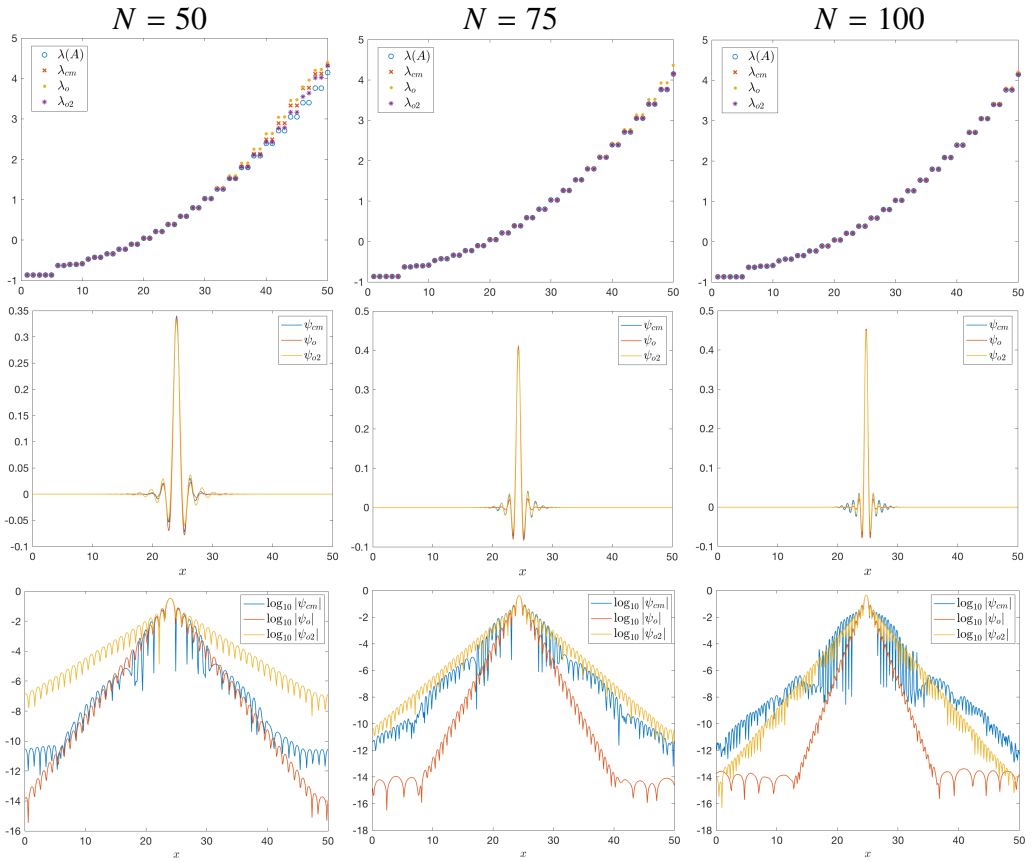



Figure 4.11: Results of problems (4.17) and (4.18) for $N = 50$(first column), $N = 75$(second column) and $N = 100$(third column). First row: the first 50 eigenvalues of $A$ and those of the compressed problems. Second row: examples of local basis functions. Third row: examples of local basis functions in log scale.

We compare the approximate eigenvalues to the first 50 eigenvalues of $A$. The first row of Figure 4.11 shows that both methods give very good approximations of $\lambda(A)$. And when $N$ increases, the approximations become better. But relatively, the results $\lambda_{cm}$ from Figure 4.11 is closer to the ground truth than our results $\lambda_o$ from (4.18). To improve our results, we simply solve problem (4.18) again, but this time using previous result $\Psi_o$ as the dual basis. That is we compute $\Psi_{o2} = A^{-1}\Psi_o(\Psi_o^T A^{-1}\Psi_o)$, and compute eigenvalues $\lambda_{o2}$ from the general eigenvalue problem $\Psi_{o2}^T A\Psi_{o2}v = \lambda\Psi_{o2}^T\Psi_{o2}v$. We can see that the approximate eigenvalues $\lambda_{o2}$ are even closer to the ground truth. An interpretation of this improvement is that if we see $\Psi_o = A^{-1}\Phi(\Phi^T A^{-1}\Phi)^{-1}$ as a transformation from $\Phi$ to $\Psi_o$, then the part $A^{-1}\Phi$ is equivalent to applying inverse power method to make $\Psi_o$ more aligned to the eigenspace of the smallest eigenvalues, while the part $(\Phi^T A^{-1}\Phi)^{-1}$ is to force $\Psi_o^T\Phi = I$ so $\Psi_o$ inherits some weakened locality from $\Phi$. So if we apply this transformation to $\Psi_o$ again to obtain $\Psi_{o2}$, $\Psi_{o2}$ will approximate the eigenspace of the smallest eigenvalues better, but with more loss of locality.

In the second row and third row of Figure 4.11, we show some examples of the local basis functions $\psi_{cm}$, $\psi_o$ and $\psi_{o2}$ (all are normalized to have unit $l_2$ norm). Interestingly, these basis functions are not just localized as expected, but indeed they have very similar profiles. One can see that for $N = 75$, the basis functions $\psi_{cm}$ and $\psi_o$ are almost identical. So it seems that in spite of how we impose locality (either the $L_1$ minimization approach, or the construction of the dual basis $\Phi$), the local behaviors of the basis functions are determined by the operator $A$ itself. We believe that this "coincidence" is governed by some intrinsic property of $A$, which may be worth further exploring and studying. If we can understand a higher level, unified mechanism that results in the locality of the basis, we may be able to extend these methods to a more general class of operators. We also observed that as $N$ goes large, $\psi_o$ and $\psi_{o2}$ become more and more localized since the support of the dual basis functions are smaller and smaller. However the locality of $\psi_{cm}$ doesn't change much as $N$ increases, since we use the same penalty parameter $\mu = 10$ for (4.17) in this experiment.

We would like to remark that, though these two problems result in local basis functions with similar profiles, problem (4.17) requires to use the split Bregman iteration to obtain the $N$ basis functions simultaneously. In our problem (4.18), since the constraints are linear and separable, the basis functions can be obtained separately and directly without iteration. Furthermore, thanks to the exponential

decay of the basis functions, each subproblem for obtaining one basis function can be restricted to a local domain without significant loss of accuracy, and the resulting local problem can be solved very efficiently. For definitions and detailed properties of these local problems for obtaining localized basis, recall Section 2.2.2. Therefore the algorithm for solving problem (4.18) can be implemented locally and in parallel.

*Chapter 5*

# CONCENTRATION OF EIGENVALUE SUMS AND GENERALIZED LIEB'S CONCAVITY THEOREM

We introduce in this chapter our concentration inequalities on partial sums of eigenvalues of random Hermitian matrices. They are extensions of the existing analogous concentration results on extreme eigenvalues. The establishment of our new concentration results is an immediate application of a generalize Lieb's concavity theorem which is also one of our main contributions.

In Section 5.1, we present our main concentration results after briefly reviewing the corresponding existing results established by Tropp. The important concept of the $k$-trace functions and its properties are introduced in Section 5.2. Our generalized Lieb's concavity results are displayed and discussed in Section 5.3, followed by their proofs in Section 5.4. Section 5.5 is dedicated to the proofs of our concentration inequalities. We provide in Section 5.6 some fundamental supporting materials and in Section 5.7 some other interesting theoretical results on the $k$-trace.

**Notation**

Throughout this chapter, we will use some notations independent of the previous chapters, as we are focusing more on theoretical derivations.

For any positive integers $n, m$, we write $\mathbb{C}^n$ for the $n$-dimensional complex vector spaces equipped with the standard $l_2$ inner products, and $\mathbb{C}^{n\times m}$ for the space of all complex matrices of size $n\times m$. Let $\mathbf{H}_n$ be the space of all $n\times n$ Hermitian matrices, $\mathbf{H}_n^+$ be the convex cone of all $n \times n$ Hermitian, positive semidefinite matrices, and $\mathbf{H}_n^{++}$ be the convex cone of all $n \times n$ Hermitian, positive definite matrices. For any matrix $A \in \mathbf{H}_n$, we denote by $\lambda_i(A)$ the $i_{\text{th}}$ largest eigenvalue of $A$. We denote the Loewner partial orders on $\mathbf{H}_n$: for any $A, B \in \mathbf{H}_n$, we write $A \succeq B$ or $B \preceq A$ if $A - B \in \mathbf{H}_n^+$, and write $A \succ B$ or $B \prec A$ if $A - B \in \mathbf{H}_n^{++}$. We write $\mathbf{0}$ for square zero matrices of suitable size according to the context, and $I_n$ for the identity matrix of size $n \times n$.

For any function $f : \mathbb{R} \to \mathbb{R}$, the extension of $f$ to a function from $\mathbf{H}_n$ to $\mathbf{H}_n$ is given

by

$$f(A) = \sum_{i=1}^{n} f(\lambda_i) u_i u_i^*, \quad A \in \mathbf{H}_n,$$

where $\lambda_1, \lambda_2, \cdots, \lambda_n$ are the eigenvalues of $A$, and $u_1, u_2, \cdots, u_n \in \mathbb{C}^n$ are the corresponding normalized eigenvectors. A function $f$ is said to be operator monotone increasing (or decreasing) if $A \geq B$ implies $f(A) \geq f(B)$ (or $f(A) \leq f(B)$); $f$ is said to be operator convex (or concave) on some convex set $S$, if

$$\tau f(A) + (1-\tau) f(B) \geq f(\tau A + (1-\tau)B) \text{ (or } \leq f(\tau A + (1-\tau)B)),$$

for any $A, B \in S$ and any $\tau \in [0,1]$. For example, the function $A \mapsto A^r$ is both operator monotone increasing and operator concave on $\mathbf{H}_n^+$ for $r \in [0,1]$ (the Loewner-Heinz theorem [72], [46], [60], see also [24]). One can find more details and properties of matrix functions in [24, 126]. For any $A \in \mathbb{C}^{n\times m}$, we denote by $\|A\|_p$ the standard Schatten $p$-norm,

$$\|A\|_p = \mathrm{Tr}[|A|^p]^{\frac{1}{p}}, \tag{5.1}$$

where $|A| = (A^*A)^{\frac{1}{2}}$. In particular, we write $\|A\| = \|A\|_\infty =$ the largest singular value of $A$.

### 5.1 Concentration of Eigenvalue Sums

In many problems, the assemble of a large complicated matrix is by sampling independent random matrices with simpler structures. To study the spectrum of the expected matrix by only evaluating the spectrum of the sample mean, we need to know how the latter deviates from the former. Therefore, we often need to estimate the spectrum of random matrices of the form $Y = \sum_{i=1}^m X^{(i)}$, where $\{X^{(i)}\}_{1\leq i\leq m}$ is a finite sequence of independent, random matrices of the same size. In particular, we consider the Hermitian case where $X^{(i)} \in \mathbf{H}_n$. An important tool to study the extreme eigenvalues of a sum of random matrices is the following master bounds by Tropp ([122, Theorem 3.6.1]).

**Proposition 5.1.1.** *For any finite sequence of independent, random matrices* $\{X^{(i)}\}_{i=1}^m \subset \mathbf{H}_n$,

$$\mathbb{E}\lambda_{\max}\Big(\sum_{i=1}^m X^{(i)}\Big) \leq \inf_{\theta>0} \frac{1}{\theta} \log \mathrm{Tr}\big[\exp\big(\sum_{i=1}^m \log \mathbb{E}\exp(\theta X^{(i)})\big)\big], \tag{5.2a}$$

$$\mathbb{E}\lambda_{\min}\Big(\sum_{i=1}^m X^{(i)}\Big) \geq \sup_{\theta<0} \frac{1}{\theta} \log \mathrm{Tr}\big[\exp\big(\sum_{i=1}^m \log \mathbb{E}\exp(\theta X^{(i)})\big)\big]. \tag{5.2b}$$

*Furthermore, for all* $t \in \mathbb{R}$,

$$\mathbb{P}\left\{\lambda_{\max}\Big(\sum_{i=1}^{m} X^{(i)}\Big) \geq t\right\} \leq \inf_{\theta>0}\ e^{-\theta t}\mathrm{Tr}\big[\exp\big(\sum_{i=1}^{m}\log\mathbb{E}\exp(\theta X^{(i)})\big)\big], \tag{5.3a}$$

$$\mathbb{P}\left\{\lambda_{\min}\Big(\sum_{i=1}^{m} X^{(i)}\Big) \leq t\right\} \leq \inf_{\theta<0}\ e^{-\theta t}\mathrm{Tr}\big[\exp\big(\sum_{i=1}^{m}\log\mathbb{E}\exp(\theta X^{(i)})\big)\big]. \tag{5.3b}$$

Tropp's proof of Proposition 5.1.1 is based on the method of matrix transform and a theorem of Lieb (the concavity of function (1.2)). The essential use of the Lieb's theorem in Tropp's argument is to establish the Jensen's inequality

$$\mathbb{E}\mathrm{Tr}\big[\exp(H+\log A)\big] \leq \mathrm{Tr}\big[\exp(H+\log\mathbb{E}A)\big],$$

for any random matrix $A \in \mathbf{H}_n^{++}$ and any fixed $H \in \mathbf{H}_n$. Using a generalized Lieb's theorem (Theorem 5.3.4), we will extend the above inequality to

$$\mathbb{E}\big(\mathrm{Tr}_k\big[\exp(H+\log A)\big]\big)^{\frac{1}{k}} \leq \big(\mathrm{Tr}_k\big[\exp(H+\log\mathbb{E}A)\big]\big)^{\frac{1}{k}}$$

and

$$\mathbb{E}\log\mathrm{Tr}_k\big[\exp(H+\log A)\big] \leq \log\mathrm{Tr}_k\big[\exp(H+\log\mathbb{E}A)\big]$$

for any $1 \leq k \leq n$. The $k$-trace function will be introduced in detail in the next section. These more general inequalities, along with the $k$-trace inequality (5.15), will help us extend Tropp's matrix master bounds on the largest (or smallest) eigenvalue to their counterparts on the sum of the $k$ largest (or smallest) eigenvalues as follows. Recall that we denote by $\lambda_i(A)$ the $i_{\text{th}}$ largest eigenvalue of any matrix $A \in \mathbf{H}_n$.

**Theorem 5.1.2.** *Given any finite sequence of independent, random matrices* $\{X^{(i)}\}_{i=1}^m \subset \mathbf{H}_n$, *let* $Y = \sum_{i=1}^m X^{(i)}$. *Then for any* $1 \leq k \leq n$,

$$\sum_{i=1}^{k}\lambda_i(\mathbb{E}Y) \leq \mathbb{E}\sum_{i=1}^{k}\lambda_i(Y) \leq \inf_{\theta>0}\ \frac{1}{\theta}\log\mathrm{Tr}_k\big[\exp\big(\sum_{i=1}^{m}\log\mathbb{E}\exp(\theta X^{(i)})\big)\big], \tag{5.4a}$$

$$\sum_{i=1}^{k}\lambda_{n-i+1}(\mathbb{E}Y) \geq \mathbb{E}\sum_{i=1}^{k}\lambda_{n-i+1}(Y) \geq \sup_{\theta<0}\ \frac{1}{\theta}\log\mathrm{Tr}_k\big[\exp\big(\sum_{i=1}^{m}\log\mathbb{E}\exp(\theta X^{(i)})\big)\big]. \tag{5.4b}$$

*Furthermore, for all* $t \in \mathbb{R}$,

$$\mathbb{P}\left\{\sum_{i=1}^{k} \lambda_i(Y) \geq t\right\} \leq \inf_{\theta>0} e^{-\frac{\theta t}{k}} \left(\mathrm{Tr}_k\left[\exp\left(\sum_{i=1}^{m} \log \mathbb{E}\exp(\theta X^{(i)})\right)\right]\right)^{\frac{1}{k}}, \tag{5.5a}$$

$$\mathbb{P}\left\{\sum_{i=1}^{k} \lambda_{n-i+1}(Y) \leq t\right\} \leq \inf_{\theta<0} e^{-\frac{\theta t}{k}} \left(\mathrm{Tr}_k\left[\exp\left(\sum_{i=1}^{m} \log \mathbb{E}\exp(\theta X^{(i)})\right)\right]\right)^{\frac{1}{k}}. \tag{5.5b}$$

The generic estimates in Theorem 5.1.2 are not user-friendly in practice, as the $k$-trace is hardly computable in general. However, we can further establish more concrete estimates for particular random matrices in this class. For example, we consider the scenario where each $X^{(i)}$ in the sum $Y = \sum_{i=1}^{m} X^{(i)}$ also satisfies $0 \leq \lambda_n(X^{(i)}) \leq \lambda_1(X^{(i)}) \leq c$ for some uniform constant $c > 0$. One of the most studied scenarios in this setting arises with an undirected, no-selfloop, randomly weighted graph $G = (V, E, W)$ of $n$ vertices. The weights $w_{ij}$ for all edges $e_{ij}, i < j$ are uniformly bounded and follow independent distributions. Then the Laplacian of such random graph is given by $L = \sum_{1\leq i<j\leq n} w_{ij} L^{(i,j)}$, where

$$L^{(i,j)} = \begin{array}{c} \\ i \\ \\ j \\ \\ \end{array} \begin{array}{c} \begin{array}{cc} i & j \end{array} \\ \begin{bmatrix} & & \\ & 1 & -1 \\ & & \\ & -1 & 1 \\ & & \end{bmatrix} \end{array}, \quad i < j,$$

is the sub-Laplacian corresponding to the edge $e_{ij}$ with unit weight. In particular, if each weight follows a Bernoulli distribution $B(1, p)$ for some uniform constant $p \in [0, 1]$, the random graph is known as the famous Erdős-Rényi model.

For these kind of problems, one may want to study the extreme eigenvalues of $\mathbb{E}Y$ but can only afford to compute the eigenvalues of random samples of $Y$, as samples of $Y$ are much sparser than $\mathbb{E}Y$ in general. Then it is crucial to know how the eigenvalues of $Y = \sum_{i=1}^{m} X^{(i)}$ deviate from the corresponding eigenvalues of $\mathbb{E}Y$. For such purposes, Tropp used the master bounds in Proposition 5.1.1 and delicate bounds for the matrix moment generating function ([122, Lemma 5.4.1]) to prove the following Chernoff-type inequalities ([122, Theorem 5.1.1]).

**Proposition 5.1.3.** *Given any finite sequence of independent, random matrices* $\{X^{(i)}\}_{i=1}^{m} \subset \mathbf{H}_n$, *let* $Y = \sum_{i=1}^{m} X^{(i)}$. *Assume that for each* $i$, $0 \leq \lambda_n(X^{(i)}) \leq$

$\lambda_1(X^{(i)}) \leq c$ *for some uniform constant* $c \geq 0$. *Then*

$$\mathbb{E}\lambda_{\max}(Y) \leq \inf_{\theta>0} \frac{e^\theta - 1}{\theta}\lambda_{\max}(\mathbb{E}Y) + \frac{c}{\theta}\log n, \tag{5.6a}$$

$$\mathbb{E}\lambda_{\min}(Y) \geq \sup_{\theta>0} \frac{1 - e^{-\theta}}{\theta}\lambda_{\min}(\mathbb{E}Y) - \frac{c}{\theta}\log n. \tag{5.6b}$$

*Furthermore, for any* $t \geq 0$,

$$\mathbb{P}\{\lambda_{\max}(Y) \geq (1+\varepsilon)\lambda_{\max}(\mathbb{E}Y)\} \leq n\left(\frac{e^\varepsilon}{(1+\varepsilon)^{1+\varepsilon}}\right)^{\lambda_{\max}(\mathbb{E}Y)/c}, \quad \varepsilon \geq 0, \tag{5.7a}$$

$$\mathbb{P}\{\lambda_{\min}(Y) \leq (1-\varepsilon)\lambda_{\min}(\mathbb{E}Y)\} \leq n\left(\frac{e^{-\varepsilon}}{(1-\varepsilon)^{1-\varepsilon}}\right)^{\lambda_{\min}(\mathbb{E}Y)/c}, \quad \varepsilon \in [0,1), \tag{5.7b}$$

In this positive semidefinite case, we can again extend the above results of Tropp to their analogs on partial sums of eigenvalues. In particular, we will use Theorem 5.1.2, the second inequality in (5.15) and bounds on matrix moment generating functions ([122, Lemma 5.4.1]) to prove the following Chernoff-type bounds.

**Theorem 5.1.4.** *Given any finite sequence of independent, random matrices* $\{X^{(i)}\}_{i=1}^m \subset \mathbf{H}_n$, *let* $Y = \sum_{i=1}^m X^{(i)}$. *Assume that for each* $i$, $0 \leq \lambda_n(X^{(i)}) \leq \lambda_1(X^{(i)}) \leq c$ *for some uniform constant* $c \geq 0$. *Then for any* $1 \leq k \leq n$, *we have expectation estimates*

$$\mathbb{E}\sum_{i=1}^k \lambda_i(Y) \leq \inf_{\theta>0} \frac{e^\theta - 1}{\theta}\sum_{i=1}^k \lambda_i(\mathbb{E}Y) + \frac{c}{\theta}\log\binom{n}{k}, \tag{5.8a}$$

$$\mathbb{E}\sum_{i=1}^k \lambda_{n-i+1}(Y) \geq \sup_{\theta>0} \frac{1 - e^{-\theta}}{\theta}\sum_{i=1}^k \lambda_{n-i+1}(\mathbb{E}Y) - \frac{c}{\theta}\log\binom{n}{k}, \tag{5.8b}$$

*and tail bounds*

$$\mathbb{P}\left\{\sum_{i=1}^k \lambda_i(Y) \geq (1+\varepsilon)\sum_{i=1}^k \lambda_i(\mathbb{E}Y)\right\} \tag{5.9a}$$
$$\leq \binom{n}{k}^{\frac{1}{k}}\left(\frac{e^\varepsilon}{(1+\varepsilon)^{1+\varepsilon}}\right)^{\frac{1}{ck}\sum_{i=1}^k \lambda_i(\mathbb{E}Y)}, \quad \varepsilon \geq 0,$$

$$\mathbb{P}\left\{\sum_{i=1}^k \lambda_{n-i+1}(Y) \leq (1-\varepsilon)\sum_{i=1}^k \lambda_{n-i+1}(\mathbb{E}Y)\right\} \tag{5.9b}$$
$$\leq \binom{n}{k}^{\frac{1}{k}}\left(\frac{e^{-\varepsilon}}{(1-\varepsilon)^{1-\varepsilon}}\right)^{\frac{1}{ck}\sum_{i=1}^k \lambda_{n-i+1}(\mathbb{E}Y)}, \quad \varepsilon \in [0,1).$$

In Tropp's results (5.6), namely the case $k = 1$, the cost of "switching" $\lambda$ and $\mathbb{E}$ is of scale $\log n$. In our estimates (5.8), the gap factor becomes $\log \binom{n}{k} \leq k \log n$ that grows only sub-linearly in $k$, which is reasonable as we are estimating the sum of the $k$ largest (or smallest) eigenvalues. We shall further compare our estimates to another related work. Tropp et al. [40] introduced a subspace argument based on Courant–Fischer characterization of eigenvalues to prove tail bounds for all eigenvalues of $Y = \sum_{i=1}^m X^{(i)}$. Though not stated in [40], the following expectation estimates for all eigenvalues can also be established using the subspace argument. Give any finite sequence of independent, random matrices $\{X^{(i)}\}_{i=1}^m$ under the same assumption as in Theorem 5.1.4, and $Y = \sum_{i=1}^m X^{(i)}$, we have for any $1 \leq k \leq n$,

$$\mathbb{E}\lambda_k(Y) \leq \inf_{\theta>0} \frac{e^\theta - 1}{\theta} \lambda_k(\mathbb{E}Y) + \frac{c}{\theta} \log(n - k + 1), \tag{5.10a}$$

$$\mathbb{E}\lambda_k(Y) \geq \sup_{\theta>0} \frac{1 - e^{-\theta}}{\theta} \lambda_k(\mathbb{E}Y) - \frac{c}{\theta} \log k. \tag{5.10b}$$

Summing (5.10a) (or (5.10b)) for the $k$ largest (or smallest) eigenvalues, we immediately obtain

$$\mathbb{E} \sum_{i=1}^k \lambda_i(Y) \leq \inf_{\theta>0} \frac{e^\theta - 1}{\theta} \sum_{i=1}^k \lambda_i(\mathbb{E}Y) + \frac{c}{\theta} \log \prod_{i=1}^k (n - i + 1), \tag{5.11a}$$

$$\mathbb{E} \sum_{i=1}^k \lambda_{n-i+1}(Y) \geq \sup_{\theta>0} \frac{1 - e^{-\theta}}{\theta} \sum_{i=1}^k \lambda_{n-i+1}(\mathbb{E}Y) - \frac{c}{\theta} \log \prod_{i=1}^k (n - i + 1). \tag{5.11b}$$

Therefore, our expectation estimates (5.8a) and (5.8b) are sharper for partial sums of eigenvalues, as $\log \binom{n}{k} < \log \prod_{i=1}^k (n - i + 1)$ for $k > 1$. In particular, if one choose $k$ to be a fixed proportion of $n$, then $\log \binom{n}{k} = O(k)$, while $\log \prod_{i=1}^k (n - i + 1) = O(k \log n)$. Our results are then better by a factor $\log n$.

At last, we remark that if we combine Theorem 5.3.1 and the subspace argument in [40], we shall be able to derive similar expectation estimates and tail bounds for the sum of arbitrary successive eigenvalues of $Y = \sum_{i=1}^m X^{(i)}$. We will leave this potential extension to future works.

## 5.2 K-trace

For any matrix $A \in \mathbb{C}^{n\times n}$ with eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_n$, we define the $k$-trace of $A$ to be

$$\mathrm{Tr}_k[A] = \sum_{1\leq i_1<i_2<\cdots<i_k\leq n} \lambda_{i_1}\lambda_{i_2}\cdots\lambda_{i_k}, \qquad 1 \leq k \leq n. \tag{5.12}$$

In particular, $\mathrm{Tr}_1[A] = \mathrm{Tr}[A]$ is the normal trace of $A$, and $\mathrm{Tr}_n[A] = \det[A]$ is the determinant of $A$. If we write $A_{(i_1\cdots i_k, i_1\cdots i_k)}$ for the $k \times k$ principal submatrix of $A$ corresponding to the indices $i_1, i_2, \cdots, i_k$, then an equivalent definition of the $k$-trace of $A$ is given by

$$\mathrm{Tr}_k[A] = \sum_{1\le i_1<i_2<\cdots<i_k\le n} \det[A_{(i_1\cdots i_k, i_1\cdots i_k)}], \qquad 1 \le k \le n. \tag{5.13}$$

Using the second definition (5.13), one can check that for any $1 \le k \le n$, the $k$-trace enjoys the cyclic invariance property like the normal trace and the determinant. That is for any $A, B \in \mathbb{C}^n$, $\mathrm{Tr}_k[AB] = \mathrm{Tr}_k[BA]$.

The motivation of studying the $k$-trace is to provide effective estimates on the sum of the k largest (or smallest) eigenvalues of, in particular, random Hermitian matrices. As we know, the sum of the k largest eigenvalues of a Hermitian matrix $A$ (as a variable) is a convex function in $A$. So if $A$ is random, we have, for example, the expectation estimate

$$\mathbb{E}\sum_{i=1}^{k}\lambda_i(A) \ge \sum_{i=1}^{k}\lambda_i(\mathbb{E}A)$$

by Jensen's inequality. Recall that $\lambda_i(A)$ denotes the $i_{\text{th}}$ largest eigenvalue of $A$. This provides a lower bound for the left hand side if we know $\mathbb{E}A$; or a way to bound the right hand side from above if we can sample $A$. However, an estimate between these two quantities in an inverse fashion is more interesting and challenging. For the $k = 1$ case, Tropp [122] related the largest eigenvalue to the trace of the exponential using the observation

$$\lambda_1(A) \le \log \mathrm{Tr}[\exp(A)] \le \lambda_1(A) + \log n, \quad A \in \mathbf{H}_n, \tag{5.14}$$

which only introduced a gap of log scale in dimension. In particular, the first inequality in (5.14) was applied to the random matrix $A$, and the second was applied to $\mathbb{E}A$. Tropp then applied the Lieb's theorem to the intermediate quantity $\mathrm{Tr}[\exp(A)]$ (more precisely, with $A = H + \log Y$ for some fixed matrix $H$ and some random matrices $Y$) to derive inverse expectation estimates and a series of matrix concentration inequalities. Inspired by Tropp's work, we will develop expectation estimates and tail bounds on the sum of the $k$ largest eigenvalues based on an analog of (5.14) that

$$\sum_{i=1}^{k}\lambda_i(A) \le \log \mathrm{Tr}_k[\exp(A)] \le \sum_{i=1}^{k}\lambda_i(A) + \log\binom{n}{k}, \quad A \in \mathbf{H}_n, \tag{5.15}$$

This is actually the starting point of this paper. Naturally, manipulating the intermediate quantity $\mathrm{Tr}_k[\exp(A)]$ in our estimates requires extending the Lieb's theorem to a general $k$-trace version. Note that the sum of the $k$ smallest eigenvalues can be handled in a similar spirit.

Apart from its particular use discussed above, the $k$-trace is of theoretical interest by itself, as it has many interpretations corresponding to different aspects of matrix theories. Writing $D(A^{(1)}, A^{(2)}, \cdots, A^{(n)})$ the mixed discriminant of any $n$ matrices $A^{(1)}, A^{(2)}, \cdots, A^{(n)} \in \mathbb{C}^{n\times n}$, we have the identity

$$\mathrm{Tr}_k[A] = \binom{n}{k} \cdot D(\underbrace{A, \cdots, A}_{k}, \underbrace{I_n, \cdots, I_n}_{n-k}).$$

Also, if we consider the $k_{\text{th}}$ exterior algebra $\wedge^k(\mathbb{C}^n)$, we can then interpret the $k$-trace of $A$ as

$$\mathrm{Tr}_k[A] = \mathrm{Tr}_{\mathcal{L}(\wedge^k(\mathbb{C}^n))}[M_0^{(k)}(A)],$$

where $\mathrm{Tr}_{\mathcal{L}(\wedge^k(\mathbb{C}^n))}$ is the normal trace on the operator space $\mathcal{L}(\wedge^k(\mathbb{C}^n))$, and $M_0^{(k)}(A) \in \mathcal{L}(\wedge^k(\mathbb{C}^n))$ is defined as $M_0^{(k)}(A)(v_1 \wedge v_2 \wedge \cdots \wedge v_k) = Av_1 \wedge Av_2 \wedge \cdots \wedge Av_k$, for any $v_1 \wedge v_2 \wedge \cdots \wedge v_k \in \wedge^k(\mathbb{C}^n)$. These two interpretations, in fact, will provide us important tools for studying the $k$-trace and proving our generalized Lieb's concavity theorems. We will discuss more on this in Section 5.6.1 and Section 5.6.2.

Throughout this work, we will be using the following nice properties of the $k$-trace.

**Proposition 5.2.1.** *For any positive integers $n, k$, $1 \le k \le n$, the $k$-trace function $\mathrm{Tr}_k[\cdot]$ satisfies the following:*

*(i) Cyclicity:* $\mathrm{Tr}_k[AB] = \mathrm{Tr}_k[BA]$, $A, B \in \mathbb{C}^{n\times n}$.

*(ii) Homogeneity:* $\mathrm{Tr}_k[\alpha A] = \alpha^k \mathrm{Tr}_k[A]$, $A \in \mathbb{C}^{n\times n}, \alpha \in \mathbb{C}$.

*(iii) Monotonicity: For any $A, B \in \mathbf{H}_n^+$, $\mathrm{Tr}_k[A] \ge \mathrm{Tr}_k[B]$, if $A \ge B$; $\mathrm{Tr}_k[A] > \mathrm{Tr}_k[B]$, if $A \succ B$. In particular, $\mathrm{Tr}_k[A] \ge 0, A \in \mathbf{H}_n^+$.*

*(iv) Concavity: The function $A \mapsto (\mathrm{Tr}_k[A])^{\frac{1}{k}}$ is concave on $\mathbf{H}_n^+$.*

*(v) Hölder's Inequality: $\mathrm{Tr}_k[|AB|^r]^{\frac{1}{r}} \le \mathrm{Tr}_k[|A|^p]^{\frac{1}{p}} \mathrm{Tr}_k[|B|^q]^{\frac{1}{q}}$, for any $r, p, q \in (0, +\infty]$, $\frac{1}{p} + \frac{1}{q} = \frac{1}{r}$, and any $A, B \in \mathbb{C}^{n\times n}$.*

*(vi) Consistency: For any* $\tilde{n}$, $k \leq \tilde{n} \leq n$, *and any* $A \in \mathbb{C}^{\tilde{n}\times\tilde{n}}$, $\mathrm{Tr}_k\left[\begin{pmatrix} A & 0 \\ 0 & 0 \end{pmatrix}_{n\times n}\right] = \mathrm{Tr}_k[A]$.

*Proof.* (i), (ii), (iii) and (vi) can be easily verified by the definitions (5.12) and (5.13). (iv) is a consequence of the general Brunn–Minkowski theorem (Corollary 5.6.3) introduced in Section 5.6.1. (v) is a direct result of expression (5.51) in Section 5.6.2. In fact, since the normal trace enjoys the Hölder's inequality, we have

$$\begin{aligned}\mathrm{Tr}_k[|AB|^r]^{\frac{1}{r}} &= \mathrm{Tr}[|\mathcal{M}_0^{(k)}(A)\mathcal{M}_0^{(k)}(B)|^r]^{\frac{1}{r}} \\ &\leq \mathrm{Tr}[|\mathcal{M}_0^{(k)}(A)|^p]^{\frac{1}{p}}\mathrm{Tr}[|\mathcal{M}_0^{(k)}(B)|^q]^{\frac{1}{q}} \\ &= \mathrm{Tr}_k[|A|^p]^{\frac{1}{p}}\mathrm{Tr}_k[|B|^q]^{\frac{1}{q}}.\end{aligned}$$

We have used multiple properties of the operator $\mathcal{M}_0^{(k)}(A)$ introduced in Section 5.6.2. □

**A Simplified Notation**

As we will be working with the general setting of $k$-trace, there is no need to specify a particular value of $k$. Therefore, we will sometimes write

$$\phi(A) = (\mathrm{Tr}_k[A])^{\frac{1}{k}}$$

for notational simplicity. Note that the function $\phi$ also satisfies (i) cyclicity, (iii) monotonicity, (v) Hölder's inequality and (vi) consistency as in Proposition 5.2.1. But now the map $A \mapsto \phi(A)$ is homogeneous of order 1 and is concave on $\mathbf{H}_n^+$. Abusing notation, we will also refer the function $\phi$ as the $k$-trace.

## 5.3 Generalized Lieb's Concavity Theorems

We present in this section the main theoretical results of this part of work.

**Theorem 5.3.1** (Generalized Lieb's Theorem)**.** *For any* $1 \leq k \leq n$ *and any* $H \in \mathbf{H}_n$, *the function*

$$A \longmapsto \left(\mathrm{Tr}_k[\exp(H + \log A)]\right)^{\frac{1}{k}} \tag{5.16}$$

*is concave on* $\mathbf{H}_n^{++}$. *Equivalently, for any* $1 \leq k \leq n$, *the function*

$$A \longmapsto \log \mathrm{Tr}_k[\exp(H + \log A)] \tag{5.17}$$

*is concave on* $\mathbf{H}_n^{++}$.

This theorem extends the Lieb's theorem ([67, Theorem 6]) from the normal trace to the $k$-traces, and hence connects it to theories of multilinear, symmetric forms of matrices. We will give a proof of this theorem in Section 5.4.1. As we will see in the proof, Theorem 5.3.1 is a joint result of the original Lieb's theorem and the Alexandrov–Fenchel inequality for mixed discriminants (Theorem 5.6.1). One can get some first ideas by looking at three extreme cases that relate to some well-known results.

- $k = 1$: The concavity of $A \longmapsto \mathrm{Tr}[\exp(H + \log A)]$ is the original Lieb's theorem.
- $k = n$: We have $(\mathrm{Tr}_n[\exp(H + \log A)])^{\frac{1}{n}} = \det[A]^{\frac{1}{n}} \cdot \exp(\frac{1}{n}\mathrm{Tr}[H])$ and $\log \mathrm{Tr}_n[\exp(H + \log A)] = \log\det[A] + \mathrm{Tr}[H]$. The concavity of $\det[A]^{\frac{1}{n}}$ or $\log\det[A]$ is known as the Brunn–Minkowski theorem [105].
- $H = \mathbf{0}$: The concavity of $\mathrm{Tr}_k[A]^{\frac{1}{k}}$, also known as the general Brunn–Minkowski theorem, is a consequence of the Alexandrov–Fenchel inequality for mixed discriminants. We will review this in Section 5.6.1.

Theorem 5.3.4 is already sufficient to yield our concentration results in Section 5.1. However, with more advanced techniques of matrix analysis, we can actually prove stronger results in the $k$-trace setting, completing the whole story of generalizing Lieb's concavity theorem.

**Lemma 5.3.2.** *For any $r \in [0, 1]$, $s \in [0, \frac{1}{r}]$ and any $K \in \mathbb{C}^{n\times n}$, the function*

$$A \longmapsto \mathrm{Tr}_k\left[(K^* A^r K)^s\right]^{\frac{1}{k}} \tag{5.18}$$

*is concave on $\mathbf{H}_n^+$.*

**Theorem 5.3.3** (Generalized Lieb's Concavity Theorem)**.** *For any $p, q \in [0, 1]$, $s \in [0, \frac{1}{p+q}]$ and any $K \in \mathbb{C}^{n\times m}$, the function*

$$(A, B) \longmapsto \mathrm{Tr}_k\left[(B^{\frac{q}{2}} K^* A^p K B^{\frac{q}{2}})^s\right]^{\frac{1}{k}} \tag{5.19}$$

*is jointly concave on $\mathbf{H}_n^+ \times \mathbf{H}_m^+$.*

**Theorem 5.3.4** (Generalized Lieb's Theorem)**.** *For any $H \in \mathbf{H}_n$ and any $\{p_j\}_{j=1}^m \subset [0, 1]$ such that $\sum_{j=1}^m p_j \leq 1$, the function*

$$(A^{(1)}, A^{(2)}, \ldots, A^{(m)}) \longmapsto \mathrm{Tr}_k\Big[\exp\big(H + \sum_{j=1}^m p_j \log A^{(j)}\big)\Big]^{\frac{1}{k}} \tag{5.20}$$

*is jointly concave on* $(\mathbf{H}_n^{++})^{\times m}$.

Lemma 5.3.2 is a $k$-trace extension of the concave part of Lemma 2.8 in [27] (see also [47, Theorem 4.1]). The latter is a consequence of Lieb's original concavity theorem. However, we will first apply the technique of operator interpolation to prove Lemma 5.3.2 independently, and then use it to derive Theorem 5.3.3 and the other results. In fact, the convexity/concavity of function (5.18) in the trace case with different ranges of $p, s$ has been used as the first step towards many consequential results on more complicated trace functions. Theorem 5.3.3 is our generalized Lieb's concavity theorem, which extends Hiai's Theorem 2.1 in [48] (see also [25, Theorem 4.4]) from trace to $k$-trace. Note that, as stated in Hiai's theorem, the concavity also holds for $-1 \leq p, q \leq 0, \frac{1}{p+q} \leq s \leq 0$. In fact, by consistency and continuity of $\phi$, it suffices to consider the case when $n = m$ and $A, B, K$ are invertible. Then, following the discussions in [26], we have that

$$\mathrm{Tr}_k \left[ (B^{\frac{q}{2}} K^* A^p K B^{\frac{q}{2}})^s \right]^{\frac{1}{k}} = \mathrm{Tr}_k \left[ (B^{-\frac{q}{2}} K^{-1} A^{-p} (K^{-1})^* B^{-\frac{q}{2}})^{-s} \right]^{\frac{1}{k}},$$

which concludes the concavity for the mirrored range of $p, q, s$. Our derivation from Lemma 5.3.2 to Theorem 5.3.3 will be a counterpart of Zhang's simple and useful variational argument in [136] for the trace case.

Theorem 5.3.4 is a generalization of Corollary 6.1 in [67] (from trace to $k$-trace). Lieb [67] proved the original trace version by checking the non-positiveness of the second-order directional derivatives (or Hessians). We will first prove Theorem 5.3.4 for $m = 1$ by applying the Lie product formula to Lemma 5.3.2 (taking $p, q \to 0, s \to +\infty$), hence providing an alternative proof of Theorem 5.3.4. We then improve the result from $m = 1$ to $m \geq 1$ using a $k$-trace version of the Araki–Lieb–Thirring inequality (Lemma 5.4.3).

The proofs of Lemma 5.3.2, Theorem 5.3.3 and Theorem 5.3.1 will be given in Section 5.4.2.

## 5.4 Proof of Concavity Theorems

We provide in this section detailed proofs of our main results in Section 5.3. We will be heavily relying on a variety of supporting materials that are presented in Section 5.6.

### 5.4.1 A First Proof by Matrix Derivative

As Theorem 5.3.4 is sufficient to lead to our concentration results, we provide in this subsection an independent proof of it. As mentioned before, this generalized Lieb's theorem is a joint result of the original Lieb's theorem and the Alexandrov–Fenchel inequality. But we will not use the Lieb's theorem directly. Instead, we will be using the following lemma, also due to Lieb [67], which is an equivalence of the Lieb's theorem. We provide the proof here only to show its connection to the Lieb's theorem.

**Lemma 5.4.1.** *Given any $A \in \mathbf{H}_n^{++}$, $C \in \mathbf{H}_n$, define*

$$T = \int_0^\infty (A+\tau I)^{-1} C (A+\tau I)^{-1} d\tau,$$

$$R = 2\int_0^\infty (A+\tau I)^{-1} C (A+\tau I)^{-1} C (A+\tau I)^{-1} d\tau,$$

*then for any $B \in \mathbf{H}_n^+$, we have*

$$\int_0^1 ds \mathrm{Tr}[TB^s TB^{1-s}] - \mathrm{Tr}[RB] \le 0. \tag{5.21}$$

*Proof.* By Lieb's theorem (Theorem 6 [67]), for any $H \in \mathbf{H}_n$, the function $g(t) = \mathrm{Tr}[\exp(H + \log(A+tC))]$ is concave. Also this function is smooth in $t$ for $t$ small enough such that $A + tC \in \mathbf{H}_n^{++}$. Thus we have $\frac{\partial^2}{\partial t^2} g(t)|_{t=0} = g''(0) \le 0$. Write $B(t) = \exp(H + \log(A+tC))$, and

$$T(t) = \int_0^\infty (A+tC+\tau I)^{-1} C (A+tC+\tau I)^{-1} d\tau,$$

$$R(t) = 2\int_0^\infty (A+tC+\tau I)^{-1} C (A+tC+\tau I)^{-1} C (A+tC+\tau I)^{-1} d\tau.$$

It is easy to check that $\frac{\partial}{\partial t}\log(A+tC) = T(t), T'(t) = -R(t)$ by formulas (5.58) and (5.59). Then using the derivative formulas (5.57), (5.58) and (5.59), we have

$$g'(t) = \int_0^1 ds \mathrm{Tr}[(B(t))^s T(t) (B(t))^{1-s}] = \mathrm{Tr}[T(t)B(t)],$$

and

$$g''(t) = \mathrm{Tr}[T'(t)B(t)] + \int_0^1 ds \mathrm{Tr}[T(t)(B(t))^s T(t)(B(t))^{1-s}].$$

For any $B \in \mathbf{H}_n^{++}$, we may choose $H = \log B - \log A$, so that $B(0) = \exp(H + \log A) = B$. And notice that $T(0) = T$, $R(0) = R$, we thus have

$$-\mathrm{Tr}[RB] + \int_0^1 ds \mathrm{Tr}[TB^s TB^{1-s}] = g''(0) \le 0.$$

The extension to $B \in \mathbf{H}_n^+$ can be done by continuity. □

We use this variant of the Lieb's theorem since it is more convenient for us to choose arbitrary $B \in \mathbf{H}_n^{++}$ in inequality (5.21). In particular, if we choose $B$ to be diagonal with diagonal entries $b_1, b_2, \cdots, b_n$, then Lemma 5.4.1 implies that

$$\sum_{i=1}^{n} R_{ii} b_i \geq \int_0^1 ds \sum_{i=1}^{n} \sum_{j=1}^{n} T_{ij} b_j^s T_{ji} b_i^{1-s}, \tag{5.22}$$

which is a critical estimate that we will be using.

We now prove a trace inequalities using Lemma 5.4.1 and the Alexandrov–Fenchel inequality Theorem 5.6.1. This inequality can be seen as a generalization of Lemma 5.4.1 from $k = 1$ to all $1 \leq k \leq n$.

**Lemma 5.4.2.** *For arbitrary $A \in \mathbf{H}_n^{++}, B \in \mathbf{H}_n^{+}, C \in \mathbf{H}_n$, let*

$$T = \int_0^\infty (A + \tau I)^{-1} C (A + \tau I)^{-1} d\tau,$$

$$R = 2 \int_0^\infty (A + \tau I)^{-1} C (A + \tau I)^{-1} C (A + \tau I)^{-1} d\tau,$$

*then we have, for all $1 \leq k \leq n$,*

$$\begin{aligned} \int_0^1 \mathrm{Tr}[\mathcal{M}_1^{(k)}(TB^s; B^s)\mathcal{M}_1^{(k)}(TB^{1-s}; B^{1-s})]\, ds - \mathrm{Tr}[\mathcal{M}_1^{(k)}(RB; B)] \\ \leq \mathrm{Tr}[\mathcal{M}_2^{(k)}(TB, TB, B)]. \end{aligned} \tag{5.23}$$

*Proof.* We first claim that we only need to consider the case when $B = \Lambda$ is a diagonal matrix with all diagonal entries $\lambda_1, \lambda_2, \cdots, \lambda_n \geq 0$. Indeed, if $B$ is not diagonal, we consider its eigenvalue decomposition $B = U\Lambda U^T$, where $U \in \mathbb{C}^{n\times n}$ is unitary, and $\Lambda$ is a diagonal matrix whose diagonal entries $\lambda_1, \lambda_2, \cdots, \lambda_n$ are the eigenvalues of $B$. Since $B \in \mathbf{H}_n^{+}$, $\lambda_1, \lambda_2, \cdots, \lambda_n \geq 0$. If we introduce $\widetilde{A} = U^T A U, \widetilde{C} = U^T C U, \widetilde{T} = U^T T U, \widetilde{R} = U^T R U$, we have

$$\widetilde{T} = \int_0^\infty (\widetilde{A} + \tau I)^{-1} \widetilde{C} (\widetilde{A} + \tau I)^{-1} d\tau,$$

$$\widetilde{R} = 2 \int_0^\infty (\widetilde{A} + \tau I)^{-1} \widetilde{C} (\widetilde{A} + \tau I)^{-1} \widetilde{C} (\widetilde{A} + \tau I)^{-1} d\tau.$$

Then using the cyclic invariance of trace and the product properties (5.47), we have, for example,

$$\begin{aligned}
&\mathrm{Tr}[\mathcal{M}_1^{(k)}(TB^s;B^s)\mathcal{M}_1^{(k)}(TB^{1-s};B^{1-s})]\\
&=\mathrm{Tr}[\mathcal{M}_1^{(k)}(UU^TTU\Lambda^sU^T;U\Lambda^sU^T)\mathcal{M}_1^{(k)}(UU^TTU\Lambda^{1-s}U^T;U\Lambda^{1-s}U^T)]\\
&=\mathrm{Tr}[\mathcal{M}_0^{(k)}(U)\mathcal{M}_1^{(k)}(\widetilde{T}\Lambda^s;\Lambda^s)\mathcal{M}_0^{(k)}(U^T)\mathcal{M}_0^{(k)}(U)\mathcal{M}_1^{(k)}(\widetilde{T}\Lambda^{1-s};\Lambda^{1-s})\mathcal{M}_0^{(k)}(U^T)]\\
&=\mathrm{Tr}[\mathcal{M}_1^{(k)}(\widetilde{T}\Lambda^s;\Lambda^s)\mathcal{M}_0^{(k)}(U^T)\mathcal{M}_0^{(k)}(U)\mathcal{M}_1^{(k)}(\widetilde{T}\Lambda^{1-s};\Lambda^{1-s})\mathcal{M}_0^{(k)}(U^T)\mathcal{M}_0^{(k)}(U)]\\
&=\mathrm{Tr}[\mathcal{M}_1^{(k)}(\widetilde{T}\Lambda^s;\Lambda^s)\mathcal{M}_1^{(k)}(\widetilde{T}\Lambda^{1-s};\Lambda^{1-s})].
\end{aligned}$$

Using the same trick to the other terms in the inequalities (5.23), one can show that (5.23) is equivalent to

$$\begin{aligned}
&\int_0^1 \mathrm{Tr}[\mathcal{M}_1^{(k)}(\widetilde{T}\Lambda^s;\Lambda^s)\mathcal{M}_1^{(k)}(\widetilde{T}\Lambda^{1-s};\Lambda^{1-s})]\,ds-\mathrm{Tr}[\mathcal{M}_1^{(k)}(\widetilde{R}\Lambda;\Lambda)]\\
&\qquad\le \mathrm{Tr}[\mathcal{M}_2^{(k)}(\widetilde{T}\Lambda,\widetilde{T}\Lambda,\Lambda)].
\end{aligned}$$

which justifies our claim. In what follows, we will still use $A, C, T, R$ for $\widetilde{A}, \widetilde{C}, \widetilde{T}, \widetilde{R}$.

We now prove (5.23) with $B = \Lambda$ being diagonal whose diagonal entries are $\lambda_1, \lambda_2, \cdots, \lambda_n \ge 0$. Using product properties (5.47) and identities in Lemma 5.6.4, we rewrite the quantity

$$\begin{aligned}
\mathcal{I} \triangleq\ &\int_0^1 ds\mathrm{Tr}[\mathcal{M}_1^{(k)}(T\Lambda^s;\Lambda^s)\mathcal{M}_1^{(k)}(T\Lambda^{1-s};\Lambda^{1-s})]-\mathrm{Tr}[\mathcal{M}_1^{(k)}(R\Lambda;\Lambda)]\\
=\ &\int_0^1 ds\left\{\mathrm{Tr}[\mathcal{M}_1^{(k)}(T\Lambda^sT\Lambda^{1-s};\Lambda)]+\mathrm{Tr}[\mathcal{M}_2^{(k)}(T\Lambda^s\Lambda^{1-s},\Lambda^sT\Lambda^{1-s};\Lambda)]\right\}\\
&-\mathrm{Tr}[\mathcal{M}_1^{(k)}(R\Lambda;\Lambda)]\\
=\ &\int_0^1 ds\left\{\sum_{i=1}^n(\sum_{j=1}^n T_{ij}\lambda_j^sT_{ji}\lambda_i^{1-s})d_i^{(n,k)}\right.\\
&\qquad\left.+\sum_{1\le i,j\le n}(T_{ii}\lambda_i\lambda_j^sT_{jj}\lambda_j^{1-s}-T_{ji}\lambda_i\lambda_i^sT_{ij}\lambda_j^{1-s})g_{ij}^{(n,k)}\right\}\\
&-\sum_{i=1}^n R_{ii}\lambda_id_i^{(n,k)}.
\end{aligned}$$

Then replacing $b_i$ by $\lambda_id_i^{(n,k)}$ in (5.22), we have by Lemma 5.4.1

$$\sum_{i=1}^n R_{ii}\lambda_id_i^{(n,k)}\ge\int_0^1 ds\sum_{i=1}^n\sum_{j=1}^n T_{ij}(\lambda_jd_j^{(n,k)})^sT_{ji}(\lambda_id_i^{(n,k)})^{1-s}.$$

Therefore we have

$$\mathcal{I} \le \int_0^1 ds\Big\{ \sum_{1\le i,j\le n} T_{ij}T_{ji}\lambda_j^s\lambda_i^{1-s}d_i^{(n,k)} + \sum_{1\le i,j\le n}(T_{ii}T_{jj}\lambda_i\lambda_j - T_{ji}T_{ij}\lambda_i^{1+s}\lambda_j^{1-s})g_{ij}^{(n,k)} - \sum_{1\le i,j\le n} T_{ij}T_{ji}(\lambda_j d_j^{(n,k)})^s(\lambda_i d_i^{(n,k)})^{1-s}\Big\}.$$

We now investigate the integrand for any $s \in [0,1]$. We have

$$\begin{aligned}
&\sum_{1\le i,j\le n} T_{ij}T_{ji}\lambda_j^s\lambda_i^{1-s}d_i^{(n,k)} + \sum_{1\le i,j\le n}(T_{ii}T_{jj}\lambda_i\lambda_j - T_{ji}T_{ij}\lambda_i^{1+s}\lambda_j^{1-s})g_{ij}^{(n,k)}\\
&- \sum_{1\le i,j\le n} T_{ij}T_{ji}(\lambda_j d_j^{(n,k)})^s(\lambda_i d_i^{(n,k)})^{1-s}\\
=\ &\sum_{i=1}^n T_{ii}^2\lambda_i d_i^{(n,k)} + \sum_{1\le i<j\le n}|T_{ij}|^2(\lambda_j^s\lambda_i^{1-s}d_i^{(n,k)} + \lambda_i^s\lambda_j^{1-s}d_j^{(n,k)})\\
&+ \sum_{1\le i,j\le n} T_{ii}T_{jj}\lambda_i\lambda_j g_{ij}^{(n,k)} - \sum_{1\le i<j\le n}|T_{ij}|^2(\lambda_i^{1+s}\lambda_j^{1-s} + \lambda_j^{1+s}\lambda_i^{1-s})g_{ij}^{(n,k)}\\
&- \sum_{i=1}^n T_{ii}^2\lambda_i d_i^{(n,k)}\\
&- \sum_{1\le i<j\le n}|T_{ij}|^2(\lambda_j^s\lambda_i^{1-s}(d_j^{(n,k)})^s(d_i^{(n,k)})^{1-s} + \lambda_i^s\lambda_j^{1-s}(d_i^{(n,k)})^s(d_j^{(n,k)})^{1-s})\\
=\ &\sum_{1\le i,j\le n} T_{ii}T_{jj}\lambda_i\lambda_j g_{ij}^{(n,k)}\\
&+ \sum_{1\le i<j\le n}|T_{ij}|^2\Big\{\lambda_j^s\lambda_i^{1-s}d_i^{(n,k)} + \lambda_i^s\lambda_j^{1-s}d_j^{(n,k)} - (\lambda_i^{1+s}\lambda_j^{1-s} + \lambda_j^{1+s}\lambda_i^{1-s})g_{ij}^{(n,k)}\\
&\qquad - \lambda_j^s\lambda_i^{1-s}(d_j^{(n,k)})^s(d_i^{(n,k)})^{1-s} - \lambda_i^s\lambda_j^{1-s}(d_i^{(n,k)})^s(d_j^{(n,k)})^{1-s}\Big\}\\
\le\ &\sum_{1\le i,j\le n} T_{ii}T_{jj}\lambda_i\lambda_j g_{ij}^{(n,k)} - 2\sum_{1\le i<j\le n}|T_{ij}|^2\lambda_i\lambda_j g_{ij}^{(n,k)}\\
=\ &\sum_{1\le i,j\le n}(T_{ii}T_{jj}\lambda_i\lambda_j - T_{ij}T_{ji}\lambda_i\lambda_j)g_{ij}^{(n,k)}.
\end{aligned}$$

We have used $g_{ij}^{(n,k)} = g_{ji}^{(n,k)}$ and $g_{ii}^{(n,k)} = 0$. The proof of the last inequality above is as follows. For any $s \in [0,1]$, we have a Hölder-type inequality for scalars:

$$(a+b)^s(c+d)^{1-s} \ge a^s c^{1-s} + b^s d^{1-s}, \quad a,b,c,d \ge 0.$$

Then using the expansion relations (5.53),

$$d_i^{(n,k)} = \lambda_j g_{ij}^{(n,k)} + g_{ij}^{(n,k+1)}, \qquad d_j^{(n,k)} = \lambda_i g_{ij}^{(n,k)} + g_{ij}^{(n,k+1)},$$

we have

$$\begin{aligned}
&\lambda_j^s\lambda_i^{1-s}d_i^{(n,k)} + \lambda_i^s\lambda_j^{1-s}d_j^{(n,k)} - (\lambda_i^{1+s}\lambda_j^{1-s} + \lambda_j^{1+s}\lambda_i^{1-s})g_{ij}^{(n,k)} \\
&\quad - \lambda_j^s\lambda_i^{1-s}(d_j^{(n,k)})^s(d_i^{(n,k)})^{1-s} - \lambda_i^s\lambda_j^{1-s}(d_i^{(n,k)})^s(d_j^{(n,k)})^{1-s} \\
\leq\ & \lambda_j^s\lambda_i^{1-s}(\lambda_j g_{ij}^{(n,k)} + g_{ij}^{(n,k+1)}) + \lambda_i^s\lambda_j^{1-s}(\lambda_i g_{ij}^{(n,k)} + g_{ij}^{(n,k+1)}) \\
&\quad - (\lambda_i^{1+s}\lambda_j^{1-s} + \lambda_j^{1+s}\lambda_i^{1-s})g_{ij}^{(n,k)} \\
&\quad - \lambda_j^s\lambda_i^{1-s}(\lambda_i^s\lambda_j^{1-s}g_{ij}^{(n,k)} + g_{ij}^{(n,k+1)}) - \lambda_i^s\lambda_j^{1-s}(\lambda_j^s\lambda_i^{1-s}g_{ij}^{(n,k)} + g_{ij}^{(n,k+1)}) \\
=\ & -2\lambda_i\lambda_j g_{ij}^{(n,k)}.
\end{aligned}$$

Finally using Lemma 5.6.4 again, we have

$$\begin{aligned}
\mathcal{I} \leq\ & \int_0^1 ds \left\{ \sum_{1\leq i,j\leq n} (T_{ii}T_{jj}\lambda_i\lambda_j - T_{ij}T_{ji}\lambda_i\lambda_j)g_{ij}^{(n,k)} \right\} \\
=\ & \mathrm{Tr}[\mathcal{M}_2^{(k)}(T\Lambda, T\Lambda, \Lambda)].
\end{aligned}$$

This completes the proof of Lemma 5.4.2. □

We are now ready to prove Theorem 5.3.1 with all established results.

***Proof of Theorem 5.3.1***. We first prove the concavity of the function

$$f_{H,k}(A) = (\mathrm{Tr}_k[\exp(H + \log A)])^{\frac{1}{k}}.$$

Notice that given any $A \in \mathbf{H}_n^{++}$ and any $C \in \mathbf{H}_n$, there exist some $\epsilon$ such that $A + tC \in \mathbf{H}_n^{++}$ for $t \in (-\epsilon, \epsilon)$, and $f_{H,k}(A + tC)$ is continuously differentiable with respect to $t$ on $(-\epsilon, \epsilon)$. In what follows, any function of $t$ is always assumed to be defined on a reasonable neighborhood of 0 (so that $A + tC \in \mathbf{H}_n^{++}$).

Then the concavity of $f_{H,k}(A)$ on $\mathbf{H}_n^{++}$ is equivalently to the statement that $\frac{\partial^2}{\partial t^2} f_{H,k}(A + tC) \leq 0|_{t=0}$ for all choices of $A \in \mathbf{H}_n^{++}, C \in \mathbf{H}_n$. Now fix a pair $A, C$, define $B(t) = \exp(H + \log(A + tC)) \in \mathbf{H}_n^{++}$ and $g(t) = \mathrm{Tr}_k[\exp(H + \log(A + tC))] = \mathrm{Tr}[\mathcal{M}_0^{(k)}(B(t))] > 0$. Since $f_{H,k}(A + tC) = g(t)^{\frac{1}{k}}$, and

$$\frac{\partial^2}{\partial t^2} f_{H,k}(A + tC) = \frac{1}{k} g(t)^{\frac{1}{k}-2}\left(g''(t)g(t) - \frac{k-1}{k}(g'(t))^2\right),$$

we then need to show that $g(0)g''(0) \leq \frac{k-1}{k}(g'(0))^2$. Using the derivative formulas (5.58) and (5.59), we have

$$\frac{\partial}{\partial t}\log(A + tC) = \int_0^\infty (A + tC + xI_n)^{-1}C(A + tC + xI_n)^{-1} \triangleq T(t),$$

$$\frac{\partial}{\partial t}T(t) = -2\int_0^\infty (A+tC+xI_n)^{-1}C(A+tC+xI_n)^{-1}C(A+tC+xI_n)^{-1} \triangleq -R(t).$$

Then using formula (5.57), we can compute the first derivative

$$\begin{aligned}
g'(t) &= \frac{\partial}{\partial t}\mathrm{Tr}[\mathcal{M}_0^{(k)}(B(t))] \\
&= \mathrm{Tr}[\mathcal{M}_1^{(k)}(B'(t); B(t))] \\
&= \mathrm{Tr}[\mathcal{M}_1^{(k)}(\int_0^1 ds B(t)^s T(t) B(t)^{1-s}; B(t))] \\
&= \int_0^1 ds \mathrm{Tr}[\mathcal{M}_1^{(k)}(B(t)^s T(t) B(t)^{1-s}; B(t)^s B(t)^{1-s})] \\
&= \int_0^1 ds \mathrm{Tr}[\mathcal{M}_0^{(k)}(B(t)^s)\mathcal{M}_1^{(k)}(T(t); I_n)\mathcal{M}_0^{(k)}(B(t)^{1-s})] \\
&= \int_0^1 ds \mathrm{Tr}[\mathcal{M}_1^{(k)}(T(t); I_n)\mathcal{M}_0^{(k)}(B(t)^{1-s})\mathcal{M}_0^{(k)}(B(t)^s)] \\
&= \mathrm{Tr}[\mathcal{M}_1^{(k)}(T(t); I_n)\mathcal{M}_0^{(k)}(B(t))].
\end{aligned}$$

We have used the fact that $\mathcal{M}_1^{(k)}(X;Y)$ is linear in $X$, and so we can pull out the integral symbol. Then the second derivative is

$$\begin{aligned}
g''(t) &= \frac{\partial}{\partial t}\mathrm{Tr}[\mathcal{M}_1^{(k)}(T(t); I_n)\mathcal{M}_0^{(k)}(B(t))] \\
&= \mathrm{Tr}[\mathcal{M}_1^{(k)}(T(t); I_n)\mathcal{M}_1^{(k)}(B'(t); B(t))] + \mathrm{Tr}[\mathcal{M}_1^{(k)}(T'(t); I_n)\mathcal{M}_0^{(k)}(B(t))] \\
&= \int_0^1 ds \mathrm{Tr}[\mathcal{M}_1^{(k)}(T(t); I_n)\mathcal{M}_0^{(k)}(B(t)^s)\mathcal{M}_1^{(k)}(T(t); I_n)\mathcal{M}_0^{(k)}(B(t)^{1-s})] \\
&\quad - \mathrm{Tr}[\mathcal{M}_1^{(k)}(R(t); I_n)\mathcal{M}_0^{(k)}(B(t))].
\end{aligned}$$

Write $T = T(0)$, $R = R(0)$ and $B = B(0)$. We then apply Lemma 5.4.2 to reach

$$\begin{aligned}
& g(0)g''(0) \\
&= \mathrm{Tr}[\mathcal{M}_0^{(k)}(B)]\Big\{\int_0^1 ds \mathrm{Tr}[\mathcal{M}_1^{(k)}(T; I_n)\mathcal{M}_0^{(k)}(B^s)\mathcal{M}_1^{(k)}(T; I_n)\mathcal{M}_0^{(k)}(B^{1-s})] \\
&\qquad\qquad - \mathrm{Tr}[\mathcal{M}_1^{(k)}(R; I_n)\mathcal{M}_0^{(k)}(B)]\Big\} \\
&= \mathrm{Tr}[\mathcal{M}_0^{(k)}(B)]\Big\{\int_0^1 ds \mathrm{Tr}[\mathcal{M}_1^{(k)}(TB^s; B^s)\mathcal{M}_1^{(k)}(TB^{1-s}; B^{1-s})] \\
&\qquad\qquad - \mathrm{Tr}[\mathcal{M}_1^{(k)}(RB; B)]\Big\} \\
&\leq \mathrm{Tr}[\mathcal{M}_0^{(k)}(B)]\mathrm{Tr}[\mathcal{M}_2^{(k)}(TB, TB, B)].
\end{aligned}$$

To continue, we use definitions (5.46), identity (5.50) and the Alexandrov–Fenchel inequality (Theorem 5.6.1) to obtain

$$\begin{aligned}
&\mathrm{Tr}[\mathcal{M}_0^{(k)}(B)]\mathrm{Tr}[\mathcal{M}_2^{(k)}(TB,TB,B)]\\
&= \frac{n!}{k!(n-k)!} D(\underbrace{B,\cdots,B}_{k},\underbrace{I_n,\cdots,I_n}_{n-k})\\
&\qquad \times \frac{n!}{(k-2)!(n-k)!} D(TB,TB,\underbrace{B\cdots,B}_{k-2},\underbrace{I_n,\cdots,I_n}_{n-k})\\
&\leq \frac{k-1}{k}\Big(\frac{n!}{(k-1)!(n-k)!} D(TB,\underbrace{B\cdots,B}_{k-1},\underbrace{I_n,\cdots,I_n}_{n-k})\Big)^2\\
&= \frac{k-1}{k}\mathrm{Tr}[\mathcal{M}_1^{(k)}(TB,B)]^2\\
&= \frac{k-1}{k}(g'(0))^2.
\end{aligned}$$

We therefore have proved that

$$g(0)g''(0) \leq \frac{k-1}{k}(g'(0))^2.$$

The concavity of $f_{H,k}(A)$ on $\mathbf{H}_n^{++}$ then follows.

Next we prove the equivalence of (i) the concavity of the functions $f_{H,k}(A)$ on $\mathbf{H}_n^{++}$ and (ii) the concavity of the functions $\tilde{f}_{H,k} = \log \mathrm{Tr}_k[\exp(H+\log A)]$ on $\mathbf{H}_n^{++}$. (i) $\Rightarrow$ (ii) is trivial. To prove (ii) $\Rightarrow$ (i), we need the following lemma.

Let $x = (x_1, x_2) \in (0,+\infty)^2$. Define $f(x) = \mathrm{Tr}_k[\exp(H + \log(x_1A_1 + x_2A_2))]$. One can easily verify that $f_{H,k}(A)$ being concave on $\mathbf{H}_n^{++}$ is equivalent to $f(x)^{\frac{1}{k}}$ being concave on $(0,+\infty)^2$ for arbitrary but fixed choice of $A_1, A_2 \in \mathbf{H}_n^{++}, H \in \mathbf{H}_n$. Similarly, $\tilde{f}_{H,k}(A)$ being concave on $\mathbf{H}_n^{++}$ is equivalent to $\log f(x)$ being concave on $(0,+\infty)^2$ for arbitrary but fixed choice of $A_1, A_2 \in \mathbf{H}_n^{++}, H \in \mathbf{H}_n$. Using the definition of the $k$-trace $\mathrm{Tr}_k$, it is easy to check that $f(x)$ is homogeneous of order $k$. By Lemma 5.6.9, we know $f(x)^{\frac{1}{k}}$ is concave if and only if $\log f(x)$ is concave. Therefore we have (i) $\Leftrightarrow$ (ii). □

### 5.4.2 Proofs of Stronger Results

We provide in this subsection the proofs of our stronger results, Theorem 5.3.3 and Theorem 5.3.1. The first step is to prove Lemma 5.3.2 using the technique of operator interpolation as in Lemma 5.6.8. The key of applying Lemma 5.6.8 is to choose some proper holomorphic function $G(z)$ and then interpolating on some power in $[0,1]$. In particular, we will perform interpolation on $w = \frac{1}{s}$ to prove Lemma 5.3.2.

Our choice of the holomorphic functions $G(z)$ in the following proof is inspired by Lieb's constructions in [67] for the use of maximum modulus principle. Recall that we will write $\phi(\cdot) = \mathrm{Tr}_k[]^{\frac{1}{k}}$ for notational simplicity.

***Proof of Lemma 5.3.2***. Note that for $s \in [0,1]$, the concavity of (5.18) is a direct consequence of the facts that (i) $\phi$ is monotone increasing and concave on $\mathbf{H}_n^+$, and (ii) $X \mapsto X^r$ and $X \mapsto X^s$ are operator monotone increasing and operator concave on $\mathbf{H}_n^+$. So in what follows we may assume that $1 \leq s \leq \frac{1}{r}$. We need to show that, for any $A, B \in \mathbf{H}_n^+$ and any $\tau \in [0,1]$,

$$\tau\phi\big((K^*A^rK)^s\big) + (1-\tau)\phi\big((K^*B^rK)^s\big) \leq \phi\big((K^*C^rK)^s\big),$$

where $C = \tau A + (1-\tau)B$. We may assume that $A, B \in \mathbf{H}_n^{++}$ and $K$ is invertible. Once this is done, the general result for $A, B \in \mathbf{H}_n^+$ and $K \in \mathbb{C}^{n\times n}$ can be obtained by continuity. Let $w = \frac{1}{s} \in [r,1]$ and $\hat{r} = rs \in [0,1]$, so $r = \hat{r}w$. Let $M = C^{\frac{r}{2}}K$, and let $M = Q|M|$ be the polar decomposition of $M$ for some unitary matrix $Q$. Since $C, K$ are both invertible, $|M| \in \mathbf{H}_n^{++}$. We then define two functions from $\mathcal{S}$ to $\mathbb{C}^{n\times n}$:

$$G_A(z) = A^{\frac{\hat{r}z}{2}}C^{-\frac{\hat{r}z}{2}}Q|M|^{\frac{z}{w}}, \quad G_B(z) = B^{\frac{\hat{r}z}{2}}C^{-\frac{\hat{r}z}{2}}Q|M|^{\frac{z}{w}}, \quad z \in \mathcal{S},$$

where $\mathcal{S}$ is given by (5.61). In what follows we will use $X$ for $A$ or $B$. We then have

$$\begin{aligned}\phi\big((K^*X^rK)^s\big) &= \phi\big((M^*C^{-\frac{r}{2}}X^rC^{-\frac{r}{2}}M)^s\big)\\ &= \phi\big((|M|Q^*C^{-\frac{\hat{r}w}{2}}X^{\frac{\hat{r}w}{2}}X^{\frac{\hat{r}w}{2}}C^{-\frac{\hat{r}w}{2}}Q|M|)^{\frac{1}{w}}\big)\\ &= \phi\big(|G_X(w)|^{\frac{2}{w}}\big).\end{aligned}$$

Since $A, B, C, M$ are now fixed matrices in $\mathbf{H}_n^{++}$, $G_A(z)$ and $G_B(z)$ are apparently holomorphic in the interior of $\mathcal{S}$ and continuous on the boundary. Also, it is easy to check that $\|G_A(z)\|$ and $\|G_B(z)\|$ are uniformly bounded on $\mathcal{S}$, since $\mathrm{Re}(z) \in [0,1]$. Therefore we can use inequality (5.65) with $\theta = w, p_\theta = \frac{2}{w}$ to obtain

$$\begin{aligned}&\phi(|G_X(w)|^{\frac{2}{w}})\\ &\quad\leq \int_{-\infty}^{+\infty} dt\Big(\frac{2(1-w)}{wp_0}\beta_{1-w}(t)\phi(|G_X(it)|^{p_0}) + \frac{2}{p_1}\beta_w(t)\phi(|G_X(1+it)|^{p_1})\Big).\end{aligned}$$

We still need to choose some $p_0, p_1 \geq 1$ satisfying $\frac{1-w}{p_0} + \frac{w}{p_1} = \frac{1}{p_w} = \frac{w}{2}$ to proceed. Note that $G_X(it) = X^{\frac{i\hat{r}t}{2}}C^{-\frac{i\hat{r}t}{2}}Q|M|^{\frac{it}{w}}$ are now unitary matrices for all $t \in \mathbb{R}$ since $X, C, |M| \in \mathbf{H}_n^{++}$, and thus $|G_X(it)|^{p_0} = I_n$ for all $p_0$. Therefore we can take $p_0 \to +\infty, p_1 = 2$ to obtain

$$\phi(|G_X(w)|^{\frac{2}{w}}) \leq \int_{-\infty}^{+\infty} dt\,\beta_w(t)\phi\big(|G_X(1+it)|^2\big).$$

Further, for each $t \in \mathbb{R}$, we have

$$\begin{aligned}
&\phi(|G_X(1+it)|^2)\\
&= \phi(G_X(1+it)^* G_X(1+it))\\
&= \phi(|M|^{\frac{(1-it)}{w}} Q^* C^{-\frac{\hat{r}(1-it)}{2}} X^{\hat{r}} C^{-\frac{\hat{r}(1+it)}{2}} Q|M|^{\frac{(1+it)}{w}})\\
&= \phi(|M|^{\frac{1}{w}} Q^* C^{-\frac{\hat{r}(1-it)}{2}} X^{\hat{r}} C^{-\frac{\hat{r}(1+it)}{2}} Q|M|^{\frac{1}{w}}),
\end{aligned}$$

where we have used the cyclicity of $\phi$ ($|M|^{\frac{it}{w}}$ is unitary) for the last equality. Therefore we have

$$\begin{aligned}
&\tau\phi(|G_A(1+it)|^2) + (1-\tau)\phi(|G_B(1+it)|^2)\\
&= \tau\phi(|M|^{\frac{1}{w}} Q^* C^{-\frac{\hat{r}(1-it)}{2}} A^{\hat{r}} C^{-\frac{\hat{r}(1+it)}{2}} Q|M|^{\frac{1}{w}})\\
&\quad + (1-\tau)\phi(|M|^{\frac{1}{w}} Q^* C^{-\frac{\hat{r}(1-it)}{2}} B^{\hat{r}} C^{-\frac{\hat{r}(1+it)}{2}} Q|M|^{\frac{1}{w}})\\
&\le \phi(|M|^{\frac{1}{w}} Q^* C^{-\frac{\hat{r}(1-it)}{2}} (\tau A^{\hat{r}} + (1-\tau)B^{\hat{r}}) C^{-\frac{\hat{r}(1+it)}{2}} Q|M|^{\frac{1}{w}})\\
&\le \phi(|M|^{\frac{1}{w}} Q^* C^{-\frac{\hat{r}(1-it)}{2}} C^{\hat{r}} C^{-\frac{\hat{r}(1+it)}{2}} Q|M|^{\frac{1}{w}})\\
&= \phi(|M|^{\frac{2}{w}})\\
&= \phi((M^*M)^{\frac{1}{w}}).
\end{aligned}$$

The first inequality above is due to the concavity of $\phi$, the second inequality is due to (i) that $\phi$ is monotone increasing on $\mathbf{H}_n^+$ and (ii) that $X \mapsto X^{\hat{r}}$ is operator concave on $\mathbf{H}_n^+$ for $\hat{r} \in (0,1]$. Finally, since $\phi((M^*M)^{\frac{1}{w}})$ is independent of $t$, and $\beta_w(t)$ is a density on $\mathbb{R}$, we obtain that

$$\begin{aligned}
&\tau\phi((K^*A^rK)^s) + (1-\tau)\phi((K^*B^rK)^s)\\
&= \tau\phi(|G_A(w)|^{\frac{2}{w}}) + (1-\tau)\phi(|G_B(w)|^{\frac{2}{w}})\\
&\le \phi((M^*M)^{\frac{1}{w}})\\
&= \phi((K^*C^rK)^s).
\end{aligned}$$

So we have proved the concavity of (5.18) on $\mathbf{H}_n^+$. □

Our next proof, using essentially Hölder's inequalities for the $k$-trace, is adapted from Zhang's proofs of Theorem 1.1 and Theorem 3.3 in [136].

***Proof of Theorem 5.3.3***. Without loss of generality, we may assume that $m = n$. Otherwise we can replace $A$ by $\begin{pmatrix} A & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}$ and $K$ by $\begin{pmatrix} K \\ \mathbf{0} \end{pmatrix}$ if $n < m$; or replace $B$

by $\begin{pmatrix} B & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}$ and $K$ by $\begin{pmatrix} K & \mathbf{0} \end{pmatrix}$ is $n > m$. By the consistency of $\phi$, these changes of variables will not affect whether the function (5.19) is jointly concave in $(A, B)$ or not. We write $X = A^{\frac{p}{2}}$ and $Y = KB^{\frac{q}{2}}$. Let $s_1 = \frac{p+q}{p}s$, $s_2 = \frac{p+q}{q}s$, so $\frac{1}{s} = \frac{1}{s_1} + \frac{1}{s_2}$. Then for any $Z \in \mathbb{C}^{n\times n}$ that is invertible, we have by Hölder's inequality ((v) in Proposition 5.2.1) that

$$\begin{aligned}
\phi\big((B^{\frac{q}{2}}K^*A^pKB^{\frac{q}{2}})^s\big) &= \phi\big(|XZZ^{-1}Y|^{2s}\big) \\
&\le \phi\big(|XZ|^{2s_1}\big)^{\frac{s}{s_1}}\phi\big(|Z^{-1}Y|^{2s_2}\big)^{\frac{s}{s_2}} \\
&\le \frac{s}{s_1}\phi\big((Z^*X^*XZ)^{s_1}\big) + \frac{s}{s_2}\phi\big((Y^*(Z^{-1})^*Z^{-1}Y)^{s_2}\big) \\
&= \frac{s}{s_1}\phi\big((Z^*X^*XZ)^{s_1}\big) + \frac{s}{s_2}\phi\big((Z^{-1}YY^*(Z^{-1})^*)^{s_2}\big).
\end{aligned}$$

We have used the fact that $\phi\big(f(|M|)\big) = \phi\big(f(|M^*|)\big)$ for any matrix $M \in \mathbb{C}^{n\times n}$ and any function $f$, since $\phi$ is only a function of eigenvalues and the spectrums of $f(|M|)$ and $f(|M^*|)$ are the same. Let $(XY)^* = Q|(XY)^*|$ be the polar decomposition of $(XY)^*$, where $Q \in \mathbb{C}^{n\times n}$ is unitary. So we have $XYQ = |(XY)^*|$. If $X$ and $Y$ are invertible, we can particularly choose $Z = YQ|(XY)^*|^{-\frac{s_1}{s_1+s_2}}$ to have

$$XZ = XYQ|(XY)^*|^{-\frac{s_1}{s_1+s_2}} = |(XY)^*|^{\frac{s_2}{s_1+s_2}},$$

$$\text{and} \quad Z^{-1}Y = |(XY)^*|^{\frac{s_1}{s_1+s_2}}Q^*,$$

which yields the equality

$$\frac{s}{s_1}\phi\big((Z^*X^*XZ)^{s_1}\big) + \frac{s}{s_2}\phi\big((Z^{-1}YY^*(Z^{-1})^*)^{s_2}\big) = \phi\big(|(XY)^*|^{\frac{2s_1s_2}{s_1+s_2}}\big) = \phi\big(|XY|^{2s}\big).$$

Now for general $X, Y$ that are not necessarily invertible, we can always find two sequences of invertible matrices $\{X_j\}_{j=1}^{+\infty}, \{Y_j\}_{j=1}^{+\infty}$ such that (i) $X_j \to X$, $Y_j \to Y$ and (ii) $X_j^*X_j \ge X^*X$, $Y_jY_j^* \ge YY^*$. Such sequences can be easily obtained by perturbing the singular values of $X$ and $Y$. For each pair of $(X_j, Y_j)$, we can find some invertible $Z_j$ so that the above equality holds. Also, for any invertible $Z$, we have $Z^*X_j^*X_jZ \ge Z^*X^*XZ$, $Z^{-1}Y_jY_j^*(Z^{-1})^* \ge Z^{-1}YY^*(Z^{-1})^*$, and thus

$$\begin{aligned}
\phi\big((Z^*X^*XZ)^{s_1}\big) &\le \phi\big((Z^*X_j^*X_jZ)^{s_1}\big), \\
\phi\big((Z^{-1}YY^*(Z^{-1})^*)^{s_2}\big) &\le \phi\big((Z^{-1}Y_jY_j^*(Z^{-1})^*)^{s_2}\big)
\end{aligned}$$

by Theorem 5.7.6, which we will prove in Section 5.7. Then we obtain a sequence of inequalities,

$$\begin{aligned}\phi(|XY|^{2s}) &\le \inf\{\frac{s}{s_1}\phi((Z^*X^*XZ)^{s_1}) + \frac{s}{s_2}\phi((Z^{-1}YY^*(Z^{-1})^*)^{s_2}) : Z \text{ invertible}\}\\ &\le \frac{s}{s_1}\phi((Z_j^*X^*XZ_j)^{s_1}) + \frac{s}{s_2}\phi((Z_j^{-1}YY^*(Z_j^{-1})^*)^{s_2})\\ &\le \frac{s}{s_1}\phi((Z_j^*X_j^*X_jZ_j)^{s_1}) + \frac{s}{s_2}\phi((Z_j^{-1}Y_jY_j^*(Z_j^{-1})^*)^{s_2})\\ &= \phi(|X_jY_j|^{2s}).\end{aligned}$$

But since $\phi(|XY|^{2s}) = \lim_{j\to+\infty}\phi(|X_jY_j|^{2s})$ by continuity, the first inequality above must be an equality. Therefore, by substituting $X = A^{\frac{p}{2}}, Y = KB^{\frac{q}{2}}$, we obtain that

$$\begin{aligned}&\phi((B^{\frac{q}{2}}K^*A^pKB^{\frac{q}{2}})^s)\\ &= \inf\{\frac{s}{s_1}\phi((Z^*A^pZ)^{s_1}) + \frac{s}{s_2}\phi((Z^{-1}KB^qK^*(Z^{-1})^*)^{s_2}) : Z \text{ invertible}\}.\end{aligned}$$

Note that $s \in [0, \frac{1}{p+q}]$ implies $s_1 \in [0, \frac{1}{p}]$, $s_2 \in [0, \frac{1}{q}]$. By Lemma 5.3.2, the map

$$(A, B) \longmapsto \frac{s}{s_1}\phi((Z^*A^pZ)^{s_1}) + \frac{s}{s_2}\phi((Z^{-1}KB^qK^*(Z^{-1})^*)^{s_2})$$

is jointly concave in $(A, B)$ for every invertible $Z$, which then implies the joint concavity of the infimum over all invertible $Z$. □

***Proof of Theorem 5.3.4 (Part I).*** We first prove the theorem for $m = 1$. Let $r = p_1 \in [0, 1]$, and $K^{(N)} = (K^{(N)})^* = \exp(\frac{1}{2N}H)$, $N \ge 1$. Then using the Lie product formula

$$\lim_{N\to+\infty}\left(\exp(\frac{1}{2N}Y)\exp(\frac{1}{N}X)\exp(\frac{1}{2N}Y)\right)^N = \exp(X+Y), \quad X, Y \in \mathbf{H}_n,$$

we have

$$\begin{aligned}&\lim_{N\to+\infty}\phi\Big(((K^{(N)})^*A^{\frac{r}{N}}K^{(N)})^N\Big)\\ &= \lim_{N\to+\infty}\phi\left(\Big(\exp(\frac{1}{2N}H)\exp(\frac{r}{N}\log A)\exp(\frac{1}{2N}H)\Big)^N\right)\\ &= \phi(\exp(H + r\log A)).\end{aligned}$$

By Theorem 5.3.3, for each $N \ge 1$, $\phi\Big(((K^{(N)})^*A^{\frac{r}{N}}K^{(N)})^N\Big)$ is concave in $A$, thus the limit function $\phi(\exp(H + r\log A))$ is also concave in $A$. □

To go from $m = 1$ to $m > 1$ in Theorem 5.3.4, we need to use the convexity of the map $A \mapsto \phi(\exp(A))$, which we will prove via the following lemmas. They are the $k$-trace extensions of the Araki–Lieb–Thirring inequality [4], the Golden–Thompson inequality and a variant of the Peierls–Bogoliubov inequality (see, e.g., [24, Theorem 2.12]).

**Lemma 5.4.3** ($k$-trace Araki–Lieb–Thirring Inequality)**.** *For any $A, B \in \mathbf{H}_n^+$, the function*

$$t \mapsto \mathrm{Tr}_k\big[(B^{\frac{t}{2}}A^tB^{\frac{t}{2}})^{\frac{1}{t}}\big]$$

*is monotone increasing on $(0, +\infty)$, that is*

$$\mathrm{Tr}_k\big[(B^{\frac{t}{2}}A^tB^{\frac{t}{2}})^{\frac{1}{t}}\big] \le \mathrm{Tr}_k\big[(B^{\frac{s}{2}}A^sB^{\frac{s}{2}})^{\frac{1}{s}}\big], \quad 0 < t \le s. \tag{5.24}$$

*Proof.* Using the definition and properties of the operator $\mathcal{M}_0^{(k)}$ in Section 5.6.2, we have that

$$\begin{aligned}\mathrm{Tr}_k\big[(B^{\frac{t}{2}}A^tB^{\frac{t}{2}})^{\frac{1}{t}}\big] &= \mathrm{Tr}\big[\mathcal{M}_0^{(k)}((B^{\frac{t}{2}}A^tB^{\frac{t}{2}})^{\frac{1}{t}})\big] \\ &= \mathrm{Tr}\big[((\mathcal{M}_0^{(k)}(B))^{\frac{t}{2}}(\mathcal{M}_0^{(k)}(A))^t(\mathcal{M}_0^{(k)}(B))^{\frac{t}{2}})^{\frac{1}{t}}\big].\end{aligned}$$

Since $A, B \in \mathbf{H}_n^+$, $\mathcal{M}_0^{(k)}(A)$ and $\mathcal{M}_0^{(k)}(B)$ are both Hermitian and positive semidefinite. Then inequality (5.24) follows immediately from the original Araki–Lieb–Thirring inequality [4] for normal trace. □

**Lemma 5.4.4** ($k$-trace Golden–Thompson Inequality)**.** *For any $A, B \in \mathbf{H}_n$,*

$$\mathrm{Tr}_k\big[\exp(A + B)\big] \le \mathrm{Tr}_k\big[\exp(A)\exp(B)\big], \tag{5.25}$$

*with equality holds if and only if $AB = BA$.*

*Proof.* We here only prove the inequality. The condition for equality will be justified in an alternative proof of this lemma in Section 5.6.2. For any $A, B \in \mathbf{H}_n$, we have

$$\begin{aligned}\mathrm{Tr}_k\big[\exp(A + B)\big] &= \lim_{m\to+\infty} \mathrm{Tr}_k\Big[\Big(\exp\big(\frac{1}{2m}B\big)\exp\big(\frac{1}{m}A\big)\exp\big(\frac{1}{2m}B\big)\Big)^m\Big] \\ &\le \mathrm{Tr}_k\big[\exp\big(\frac{1}{2}B\big)\exp(A)\exp\big(\frac{1}{2}B\big)\big] \\ &= \mathrm{Tr}_k\big[\exp(A)\exp(B)\big].\end{aligned}$$

The first equality above is the Lie product formula, and the inequality is due to Lemma 5.4.3. □

**Lemma 5.4.5** ($k$-trace Peierls–Bogoliubov Inequality)**.** *The function*

$$A \longmapsto \log \mathrm{Tr}_k\big[\exp(A)\big] \tag{5.26}$$

*is convex on* $\mathbf{H}_n$.

*Proof.* For any $A, B \in \mathbf{H}_n$, $\tau \in (0,1)$, by Lemma 5.4.4 we have

$$\begin{aligned}\mathrm{Tr}_k\big[\exp(\tau A + (1-\tau)B)\big] &\le \mathrm{Tr}_k\big[\exp(\tau A)\exp((1-\tau)B)\big] \\ &\le \mathrm{Tr}_k\big[\exp(A)\big]^{\tau}\mathrm{Tr}_k\big[\exp(B)\big]^{1-\tau}.\end{aligned}$$

The second inequality above is Hölder's. Therefore

$$\log \mathrm{Tr}_k\big[\exp(\tau A + (1-\tau)B)\big] \le \tau \log \mathrm{Tr}_k\big[\exp(A)\big] + (1-\tau)\log \mathrm{Tr}_k\big[\exp(B)\big].$$

□

We remark that Lemma 5.4.5 can also be proved using the operator interpolation in Lemma 5.6.8. Lemma 5.4.5 immediately implies that $A \mapsto \log \phi(\exp(A)) = \frac{1}{k}\log \mathrm{Tr}_k\big[\exp(A)\big]$ is convex, and thus $A \mapsto \phi(\exp(A))$ is convex. This will help us prove improve from $m = 1$ to $m \ge 1$ in Theorem 5.3.4.

***Proof of Theorem 5.3.4 (Part II).*** Given any $\{A^{(j)}\}_{j=1}^m, \{B^{(j)}\}_{j=1}^m \subset \mathbf{H}_n^{++}$, and any $\tau \in [0,1]$, let $C^{(j)} = \tau A^{(j)} + (1-\tau)B^{(j)}$, $1 \le j \le m$. Since the map $X \mapsto \phi(\exp(X))$ is convex on $\mathbf{H}_n$, the map $X \mapsto \phi(\exp(L+X))$ is also convex on $\mathbf{H}_n$ for arbitrary $L \in \mathbf{H}_n$. Now define

$$L = H + \sum_{j=1}^m p_j \log C^{(j)}, \quad r = \sum_{j=1}^m p_j \le 1.$$

If $r = 0$, the result is trivial; so we may assume that $r > 0$. We then have that

$$\begin{aligned}&\phi\big(\exp(H + \sum_{j=1}^m p_j \log X^{(j)})\big) \\ &= \phi\Big(\exp\big(H + r\sum_{j=1}^m \frac{p_j}{r}(\log X^{(j)} - \log C^{(j)}) + \sum_{j=1}^m p_j \log C^{(j)}\big)\Big) \\ &= \phi\Big(\exp\big(L + r\sum_{j=1}^m \frac{p_j}{r}(\log X^{(j)} - \log C^{(j)})\big)\Big) \\ &\le \sum_{j=1}^m \frac{p_j}{r}\phi\big(\exp(L + r\log X^{(j)} - r\log C^{(j)})\big), \quad X^{(j)} = A^{(j)}, B^{(j)}.\end{aligned}$$

For each $j$, by the concavity of (5.20) for $m = 1$, we have

$$\begin{aligned}
&\tau\phi(\exp(L + r\log A^{(j)} - r\log C^{(j)})) \\
&\quad + (1-\tau)\phi(\exp(L + r\log B^{(j)} - r\log C^{(j)})) \\
&\le \phi(\exp(L + r\log(\tau A^{(j)} + (1-\tau)B^{(j)}) - r\log C^{(j)})) \\
&= \phi(\exp(L)).
\end{aligned}$$

Therefore we obtain that

$$\begin{aligned}
&\tau\phi(\exp(H + \sum_{j=1}^{m} p_j \log A^{(j)})) + (1-\tau)\phi(\exp(H + \sum_{j=1}^{m} p_j \log B^{(j)})) \\
&\le \sum_{j=1}^{m} \frac{p_j}{r}\phi(\exp(L)) \\
&= \phi(\exp(H + \sum_{j=1}^{m} p_j \log C^{(j)})),
\end{aligned}$$

that is, (5.20) is jointly concave on $(\mathbf{H}_n^{++})^{\times m}$ for all $m \ge 1$. □

### 5.4.3 Revisiting Previous Proofs in the Trace Case

In this section, we will review some previous works on concavity results of trace functions. The purpose is to compare by example the spirits of methods from different perspectives, so as to explain why we have chosen the interpolation technique by Stein and the variational method by Zhang to prove our main results. For a whole story of known results on both convexity and concavity, one may refer to [3, 24, 26, 67, 136]. As mentioned in the introduction, many alternative proofs of Lieb's concavity theorem (the concavity of function (1.3)) have been found since its original establishment by Lieb in 1973. A proof using matrix tensors was given by Ando [3] in 1979 (see also Carlen [24]). Ando interpreted $\mathrm{Tr}[K^* A^p K B^q]$ as an inner product on the tensor space $\mathbb{C}^n \otimes \mathbb{C}^m$ and translated the Lieb's concavity theorem to the statement that the map $(A, B) \mapsto A^p \otimes B^q$ is operator concave. Ando then proved the latter using the integral representation of $A^p$ (see below). Here $\otimes$ is the Kronecker product. Later, Nikoufar et al. [85] provided a simpler proof for the concavity of $(A, B) \mapsto A^p \otimes B^q$ using the concept of matrix perspectives (see, e.g., [36]). We summarize the ideas of their proofs as follows. For simplicity, we assume that $p + q = 1$. The result for $p + q = r < 1$ can be further obtained by using the fact that $A \mapsto A^r$ is operator monotone increasing and operator concave for $r \in [0, 1]$. For $p \in [0, 1]$, the map $A \mapsto (A \otimes I_m)^p = A^p \otimes I_m$ from $\mathbf{H}_n^+$ to $\mathbf{H}_{nm}^+$ is

operator concave, and thus its perspective from $\mathbf{H}_n^+ \times \mathbf{H}_m^+$ to $\mathbf{H}_{nm}^+$,

$$(A, B) \mapsto (I_n \otimes B)^{\frac{1}{2}}\big((I_n \otimes B)^{-\frac{1}{2}}(A \otimes I_m)(I_n \otimes B)^{-\frac{1}{2}}\big)^p (I_n \otimes B)^{\frac{1}{2}} = A^p \otimes B^{1-p},$$

is jointly operator concave in $(A, B)$. The simplified expression above results from the fact that $A \otimes I_m$ commutes with $I_n \otimes B$. For any $K \in \mathbb{C}^{n\times m}$, we have the identity (a variant of Ando's interpretation)

$$\mathrm{Tr}[K^* A^p K B^{1-p}] = \Big\langle \sum_{j=1}^m (K e_j^{(m)}) \otimes e_j^{(m)}, A^p \otimes (B^T)^{1-p} \sum_{j=1}^m (K e_j^{(m)}) \otimes e_j^{(m)} \Big\rangle_{\mathbb{C}^n \otimes \mathbb{C}^m}, \tag{5.27}$$

where $B^T$ is the transpose of $B$, and $e_j^{(m)} = (0, \ldots, \overset{j_{\text{th}}}{1}, \ldots, 0) \in \mathbb{C}^m$. Note that since $B \in \mathbf{H}_n^+$, $B^T$ is also in $\mathbf{H}_n^+$. Since $B \mapsto B^T$ is linear, the joint operator concavity of $(A, B) \mapsto A^p \otimes B^{1-p}$ then implies the joint concavity of $(A, B) \mapsto \mathrm{Tr}[K^* A^p K B^{1-p}]$.

As an application, Carlen and Lieb [27] applied the Lieb's concavity theorem to prove the concavity of $A \mapsto \mathrm{Tr}[(K^* A^r K)^s]$ for $r \in [0, 1], s \in [1, \frac{1}{r}]$ (they used a slightly different but equivalent expression) based on a variational characterization of this function (the supremum part of [27, Lemma 2.2]). We here provide a simplified proof that captures the main spirit. For any $A, B \in \mathbf{H}_n^+$, $K \in \mathbb{C}^{n\times n}$, let

$$X = (K^* A^r K)^s, \quad Y = (K^* B^r K)^s.$$

Then for any $\tau \in [0, 1]$, note that $\frac{1}{s} \le 1, r + (1 - \frac{1}{s}) \le 1$, we have

$$\begin{aligned}
&\tau \mathrm{Tr}[X] + (1-\tau)\mathrm{Tr}[Y] \\
&= \tau \mathrm{Tr}\big[K^* A^r K X^{1-\frac{1}{s}}\big] + (1-\tau)\mathrm{Tr}\big[K^* B^r K Y^{1-\frac{1}{s}}\big] \\
&\le \mathrm{Tr}\big[K^* (\tau A + (1-\tau)B)^r K (\tau X + (1-\tau)Y)^{1-\frac{1}{s}}\big] \\
&\le \mathrm{Tr}\big[(K^* (\tau A + (1-\tau)B)^r K)^s\big]^{\frac{1}{s}} \mathrm{Tr}\big[\tau X + (1-\tau)Y\big]^{1-\frac{1}{s}} \\
&= \mathrm{Tr}\big[(K^* (\tau A + (1-\tau)B)^r K)^s\big]^{\frac{1}{s}} \big(\tau \mathrm{Tr}[X] + (1-\tau)\mathrm{Tr}[Y]\big)^{1-\frac{1}{s}},
\end{aligned} \tag{5.28}$$

where the first inequality is due to Lieb's concavity theorem with $p = r, q = 1 - \frac{1}{s}, p + q = r + 1 - \frac{1}{s} \le 1$, and the second inequality is Hölder's. The above then simplifies to

$$\tau \mathrm{Tr}\big[(K^* A^r K)^s\big] + (1-\tau)\mathrm{Tr}\big[(K^* B^r K)^s\big] \le \mathrm{Tr}\big[(K^* (\tau A + (1-\tau)B)^r K)^s\big],$$

which concludes the concavity of $A \mapsto \mathrm{Tr}[(K^* A^r K)^s]$.

The variational characterizations of $\mathrm{Tr}[(K^*A^rK)^s]$ in [27] can be abstracted to the following two formulas ([26, Lemma 12]): for any $X \in \mathbf{H}_n^+$,

$$\mathrm{Tr}[X^s] = \sup\left\{s\mathrm{Tr}[XY] - (s-1)\mathrm{Tr}[Y^{\frac{s}{s-1}}] : \ Y \in \mathbf{H}_n^+\right\}, \quad \text{if } s > 1 \text{ or } s < 0, \tag{5.29}$$

$$\mathrm{Tr}[X^s] = \inf\left\{s\mathrm{Tr}[XY] + (1-s)\mathrm{Tr}[Y^{-\frac{s}{1-s}}] : \ Y \in \mathbf{H}_n^{++}\right\}, \quad \text{if } 0 < s < 1. \tag{5.30}$$

Further, these variational formulas were used to derive the convexity/concavity of the function $(A, B) \mapsto \mathrm{Tr}[(B^{\frac{q}{2}}K^*A^pKB^{\frac{q}{2}})^s]$ for a partial range of $p, q, s$ (partial to the necessary conditions on $p, q, s$ for the corresponding convexity/concavity to hold). For example, formula (5.30) was used by Carlen et al. to prove the concavity for $0 \le p, q \le 1, 0 < s \le \frac{1}{1+q}$ [25, Theorem 4.4]. Recently, the above formulas were modified by Zhang to the following [136, Theorem 3.3]: for any $X, Y \in \mathbb{C}^{n\times n}$ and any $r_0, r_1, r_2 > 0$ such that $\frac{1}{r_0} = \frac{1}{r_1} + \frac{1}{r_2}$,

$$\mathrm{Tr}[|XY|^{r_1}] = \sup\left\{\frac{r_1}{r_0}\mathrm{Tr}[|XZ|^{r_0}] - \frac{r_1}{r_2}\mathrm{Tr}[|Y^{-1}Z|^{r_2}] : \ Z \in \mathbb{C}^{n\times n}\right\}, \tag{5.31}$$

$$\mathrm{Tr}[|XY|^{r_0}] = \inf\left\{\frac{r_0}{r_1}\mathrm{Tr}[|XZ|^{r_1}] + \frac{r_0}{r_2}\mathrm{Tr}[|Z^{-1}Y|^{r_2}] : \ Z \in \mathbb{C}^{n\times n} \text{ invertible}\right\}. \tag{5.32}$$

Zhang then used them to provide a unified variational proof of the joint convexity/concavity of $(A, B) \mapsto \mathrm{Tr}[(B^{\frac{q}{2}}K^*A^pKB^{\frac{q}{2}})^s]$ for the full range of $p, q, s$, finally confirming that the sufficient conditions on $p, q, s$ coincide with the necessary conditions.

These arguments using matrix tensors and variational forms were also adopted by Tropp [122] to provide an alternative proof of the concavity of $A \mapsto \mathrm{Tr}[\exp(H + \log A)]$. Tropp's proof is based on his variational formula for trace,

$$\mathrm{Tr}[M] = \sup_{T\in\mathbf{H}_n^{++}} \mathrm{Tr}\big[T\log M - T\log T + T\big], \quad M \in \mathbf{H}_n^{++}, \tag{5.33}$$

which relies on the non-negativeness of the matrix relative entropy

$$D(T; M) = \mathrm{Tr}\big[T(\log T - \log M) - (T - M)\big], \quad T, M \in \mathbf{H}_n^{++}.$$

The non-negativeness of $D(T; M)$ is a classical result of Klein's inequality (see Petz [93, Proposition 3], Carlen [24, Theorem 2.11] or Tropp [122, Proposition 8.3.5]). Tropp substituted $M = \exp(H + \log A)$ in (5.33) to obtain

$$\mathrm{Tr}[\exp(H + \log A)] = \sup_{T\in\mathbf{H}_n^{++}} \big(\mathrm{Tr}[TH] + \mathrm{Tr}[A] - D(T; A)\big).$$

The concavity of $A \mapsto \mathrm{Tr}[\exp(H + \log A)]$ then follows from this variational expression, the joint convexity of $D(T; A)$ in $(T, A)$, and the fact that $g(x) = \sup_{y\in\Omega} f(x, y)$

is concave in $x$ if $f(x, y)$ is jointly concave in $(x, y)$ and $\Omega$ is convex (see, e.g., [27, Lemma 2.3]). The joint convexity of the relative entropy $D(T; A)$ was first due to Lindblad [70]. One can also see Ando [3], Carlen [24] and Tropp [122] for alternative proofs.

A Methodology of another flavor arose from the use of complex analysis. In the same year of Lieb's original paper on his concavity theorem, Epstein [37] provided a unified way of proving the concavity of $A \mapsto \mathrm{Tr}[K^*A^pKA^q]$, $A \mapsto \mathrm{Tr}[(K^*A^sK)^{\frac{1}{s}}]$ and $A \mapsto \mathrm{Tr}[\exp(H + \log A)]$, using a derivative argument based on the theory of Herglotz functions (functions that are analytic in the open upper half plane $\mathbb{C}_+$ and have a positive imaginary part). Epstein's method relies on integral representations of matrix powers, which has a deep connection to a profound theorem of Loewner's : a real-valued function $f$ on $(0, +\infty)$ is operator monotone if and only if it admits an analytic continuation to a Herglotz function. Loewner's theorem provides a convenient tool for understanding trace functions by passing the study of a desired property from the integral to the integrand that has a relatively simpler form. One may see the book of Donoghue [35] for a full account of this theory. Epstein's approach was further developed by Hiai [47, 48] and, for example, adopted in his first proof of the concavity of $(A, B) \mapsto \mathrm{Tr}[(B^{\frac{q}{2}}K^*A^pKB^{\frac{q}{2}})^s]$ for the full range $0 \le p, q \le 1, 0 \le s \le \frac{1}{p+q}$ [48, Theorem 2.1]. Specifically, by substituting $\sigma = s(p + q) < 1$, $X = (B^{\frac{q}{2}}K^*A^pKB^{\frac{q}{2}})^{\frac{1}{p+q}}$ in the integral formula,

$$X^\sigma = \frac{\sin(\pi\sigma)}{\pi}\int_0^\infty t^{-1+\sigma}(I_n + tX^{-1})^{-1}dt, \quad 0 < \sigma < 1,\ X \in \mathbf{H}_n^{++}, \tag{5.34}$$

Hiai passed the joint concavity of $(A, B) \mapsto \mathrm{Tr}[(B^{\frac{q}{2}}K^*A^pKB^{\frac{q}{2}})^s]$ to the joint concavity of $(A, B) \mapsto \mathrm{Tr}\Big[1 + t(B^{\frac{q}{2}}K^*A^pKB^{\frac{q}{2}})^{-\frac{1}{p+q}}\Big]^{-1}$ (the case $s(p + q) = 1$ was handled differently by directly taking $t \to +\infty$). He then proved the latter using the derivative argument introduced by Epstein.

The introduction of complex analysis into the field has also led to another branch of methods based on interpolation theories. In his original proof of the concavity of $(A, B) \mapsto \mathrm{Tr}[K^*A^pKB^q]$ for $0 \le p, q \le 1, p + q \le 1$ [67, Theorem 1], Lieb made use of the maximum modulus principle to concentrate the powers of $A^p, B^q$ to the only power $p + q$ on $A$ or $B$, and then proceeded with the operator concavity of $X \mapsto X^{p+q}$. This technique, relying on the holomorphicity of $X^z$ ($X \in \mathbf{H}_n^+$) as a function of $z$, already shed some light on the use of complex interpolation theories. Later, Uhlmann [124] applied interpolation theories explicitly to again prove Lieb's concavity theorem, by interpreting $\mathrm{Tr}[K^*A^pKB^{1-p}]$ as an interpolation be-

tween $\mathrm{Tr}[K^*AK]$ and $\mathrm{Tr}[K^*KB]$. Uhlmann's quadratic interpolation of seminorms extended the relevant works of Lieb to positive linear forms of arbitrary *-algebras. Kosaki [58] further explored the idea of quadratic interpolation of seminorms and captured Lieb's concavity theorem in the frame of general interpolation theories.

The above methodologies all have a unique perspective in understanding complicated trace functions. However, some of them are found hardly generalizable to $k$-trace, as they more or less rely on the linearity of the normal trace. For example, Ando's identity (5.27) and Tropp's variational formula (5.33). Our $k$-trace function $\phi$ for $k > 1$ is at best sub-additive since it is concave and homogeneous of order 1. Hiai's use of the integral formula also lives on linearity in that the trace operation can be pulled into the integral. Though we can interpret the $k$-trace as the trace of an anti-symmetric tensor, the power of $\frac{1}{k}$ will have to stay out of the integral, and the anti-symmetric tensor in the integrand will also bring huge difficulties to the derivative argument that comes after.

The variational method introduced by Carlen and Lieb needs the linearity of trace as well in proving the concavity of $A \mapsto \mathrm{Tr}[(K^*A^rK)^s]$. One can see this in the last equality of (5.28), which will become an inequality in the undesirable direction if $\mathrm{Tr}[\cdot]$ is replaced by $\mathrm{Tr}_k[\cdot]^{\frac{1}{k}}$, since $\mathrm{Tr}_k[X]^{\frac{1}{k}}$ is concave in $X$. In fact, the argument in (5.28) that reduces the concavity of $A \mapsto \mathrm{Tr}[(K^*A^rK)^s]$ for $r \in [0,1], s \in [1, \frac{1}{r}]$ to the joint concavity of $(A,B) \mapsto \mathrm{Tr}[K^*A^pKB^q]$ for $p, q \in [0,1], p+q \leq 1$ was performed in a variational manner in [27] based on the following formula (which is equivalent to the expression of the supremum case in [27, Lemma 2.2]):

$$\mathrm{Tr}[(K^*A^rK)^s] = \sup_{X \in \mathbf{H}_n^{++}} \left\{ s\mathrm{Tr}[K^*A^rKX^{1-\frac{1}{s}}] - (s-1)\mathrm{Tr}[X] \right\} = \sup_{X \in \mathbf{H}_n^{++}} \Psi(A,X), \tag{5.35}$$

which can be derived using Hölder's inequality for trace. Recall that $g(x) = \sup_{y\in\Omega} f(x,y)$ is concave in $x$ if $f(x,y)$ is jointly concave in $(x,y)$ and $\Omega$ is convex. Since $\mathrm{Tr}[K^*A^rKX^{1-\frac{1}{s}}]$ is jointly concave in $(A,X)$ (as $r + (1-\frac{1}{s}) \leq 1$) and $\mathrm{Tr}[X]$ is linear in $X$, the function $\Psi(A,X)$ is jointly concave in $(A,X)$. The concavity of $A \mapsto \mathrm{Tr}[(K^*A^rK)^s]$ then follows from (5.35). It is then natural to consider a similar variational formula for the $k$-trace that can also be shown by Hölder's inequality:

$$\begin{aligned} \mathrm{Tr}_k[(K^*A^rK)^s]^{\frac{1}{k}} &= \sup_{X \in \mathbf{H}_n^{++}} \left\{ s\mathrm{Tr}_k[K^*A^rKX^{1-\frac{1}{s}}]^{\frac{1}{k}} - (s-1)\mathrm{Tr}_k[X]^{\frac{1}{k}} \right\} \\ &= \sup_{X \in \mathbf{H}_n^{++}} \Psi_k(A,X). \end{aligned} \tag{5.36}$$

Note that for $s > 1$, we have $-(s-1) < 0$ and thus $-(s-1)\mathrm{Tr}_k[X]^{\frac{1}{k}}$ is convex in $X$ since $\mathrm{Tr}_k[\cdot]^{\frac{1}{k}}$ is concave (this sign of $-(s-1)$ does not give trouble in the trace case due to the linearity of trace). Therefore, even provided that $(A, X) \mapsto \mathrm{Tr}_k[K^*A^rKX^{1-\frac{1}{s}}]^{\frac{1}{k}}$ is jointly concave, the function $\Psi_k(A, X)$ is not guaranteed to be jointly concave in $(A, X)$, and the variational formula in (5.36) fails to conclude the concavity of $A \mapsto \mathrm{Tr}_k[(K^*A^rK)^s]^{\frac{1}{k}}$. As a consequence, we have not found a way to adapt this particular argument into a proof of Lemma 5.3.2.

However, these variational approaches, especially Zhang's variational characterizations, are conveniently applicable to the derivation of the more general cases given Lemma 5.3.2, as we have seen in the proof of Theorem 5.3.3. One can see that the $k$-trace version of Hölder's inequality plays an essential role in the process, which has suggested us to employ complex interpolation theories in the first place as interpolation of operators is based essentially on Hölder's inequality. In particular, we found the operator interpolation technique (Theorem 5.6.7) developed by Stein [116](1956) nicely compatible to our problem. One can derive a variety of interpolation inequalities systematically by choosing $G(z)$ properly in inequality (5.62). The choice of $G(z)$ we make, inspired by Lieb's original construction, gives the proof of our Lemma 5.3.2. We remark that, Lemma 5.6.8 does not only apply to trace or $k$-trace, but also to any continuous matrix function $\phi : \mathbf{H}_n^+ \to [0, +\infty)$ that satisfies Hölder's inequality and is unitary invariant in the sense that $\phi(U^*XU) = \phi(X)$ for arbitrary $X \in \mathbf{H}_n^+$ and $U \in \mathbb{C}^{n\times n}$ unitary. In fact, these two properties suffice to imply that $\log \circ \phi \circ \exp$ is convex on $\mathbf{H}_n$, which, along with a canonical majorization argument, will yield the inequality (5.63) (see, e.g., [50]).

## 5.5 From Lieb's Theorem to Concentration

Tropp's proof of the master bounds relies on a critical use of the Lieb's theorem. To be specific, Tropp used the concavity of $A \mapsto \mathrm{Tr}[\exp(H + \log A)]$ to prove the subadditivity of matrix cumulant generating function. For the sequence $\{X^{(i)}\}_{i=1}^m \subset \mathbf{H}_n$ under the same setting,

$$\mathbb{E}\mathrm{Tr}[\exp(\sum_{i=1}^m X^{(i)})] \leq \mathrm{Tr}[\exp(\sum_{i=1}^m \log \mathbb{E}\exp X^{(i)})]. \tag{5.37}$$

Similarly, to prove Theorem 5.1.2, we need to extend (5.37) to the following lemma using our generalized Lieb's theorem.

**Lemma 5.5.1.** *Let $A^{(1)}, A^{(2)}, \cdots, A^{(m)} \in \mathbf{H}_n^{++}$ be m independent, random, positive*

*definite matrices. Then we have for any* $1 \leq k \leq n$,

$$\mathbb{E}(\mathrm{Tr}_k[\exp(\sum_{i=1}^m \log A^{(i)})])^{\frac{1}{k}} \leq (\mathrm{Tr}_k[\exp(\sum_{i=1}^m \log \mathbb{E}A^{(i)})])^{\frac{1}{k}}, \tag{5.38a}$$

$$\mathbb{E}\log \mathrm{Tr}_k[\exp(\sum_{i=1}^m \log A^{(i)})] \leq \log \mathrm{Tr}_k[\exp(\sum_{i=1}^m \log \mathbb{E}A^{(i)})]. \tag{5.38b}$$

*Proof.* We here only prove (5.38a). The proof for (5.38b) is similar. Since each random matrix $A^{(i)}$ always lies in $\mathbf{H}_n^{++}$, we may apply Theorem 5.3.1 to get a Jensen's inequality

$$\mathbb{E}(\mathrm{Tr}_k[\exp(H + \log A^{(i)})])^{\frac{1}{k}} \leq (\mathrm{Tr}_k[\exp(H + \log \mathbb{E}A^{(i)})])^{\frac{1}{k}},$$

for arbitrary $H \in \mathbf{H}_n$. And since $A^{(1)}, A^{(2)}, \cdots, A^{(m)}$ are independent, we can split the expectation $\mathbb{E}$ into $\mathbb{E}_1\mathbb{E}_2\cdots\mathbb{E}_m$, where $\mathbb{E}_i$ is the expectation operator with respect to the random matrix $A^{(i)}$. So then we may apply Theorem 5.3.1 repeatedly to get

$$\begin{aligned}
&\mathbb{E}(\mathrm{Tr}_k[\exp(\sum_{i=1}^m \log A^{(i)})])^{\frac{1}{k}} \\
=\ &\mathbb{E}_1\cdots\mathbb{E}_m(\mathrm{Tr}_k[\exp(\sum_{i=1}^{m-1} \log A^{(i)} + \log A^{(m)})])^{\frac{1}{k}} \\
\leq\ &\mathbb{E}_1\cdots\mathbb{E}_{m-1}(\mathrm{Tr}_k[\exp(\sum_{i=1}^{m-1} \log A^{(i)} + \log \mathbb{E}_m A^{(m)})])^{\frac{1}{k}} \\
&\cdots \\
\leq\ &(\mathrm{Tr}_k[\exp(\sum_{i=1}^m \log \mathbb{E}A^{(i)})])^{\frac{1}{k}}.
\end{aligned}$$

□

Our proof of Theorem 5.1.2 will basically follow Tropp's proof of Theorem 3.6.1 in [122], but with the normal trace Tr replaced by the general $k$-trace $\mathrm{Tr}_k$ for $1 \leq k \leq n$.

***Proof of Theorem 5.1.2***. We first prove (5.4a) and (5.5a). The first inequality in (5.4a) is trivial, because the sum of a Hermitian matrix's $k$ largest eigenvalues is a convex function of the matrix itself. Indeed we have

$$\mathbb{E}\sum_{i=1}^k \lambda_i(Y) = \mathbb{E}\sup_{\substack{Q\in\mathbb{C}^{n\times k}\\ Q^*Q=I_k}} \mathrm{Tr}[Q^*YQ] \geq \sup_{\substack{Q\in\mathbb{C}^{n\times k}\\ Q^*Q=I_k}} \mathrm{Tr}[Q^*(\mathbb{E}Y)Q] = \sum_{i=1}^k \lambda_i(\mathbb{E}Y).$$

For the second inequality in (5.4a), we apply a similar technique, the matrix Laplace transform, as in [122]. The difference is that in the first step, we do not switch the expectation operator and the logarithm. For any $\theta > 0$, we have

$$\begin{aligned}\mathbb{E}\sum_{i=1}^{k}\lambda_i(Y) &= \frac{1}{\theta}\mathbb{E}\log\exp\big(\sum_{i=1}^{k}\lambda_i(\theta Y)\big)\\ &= \frac{1}{\theta}\mathbb{E}\log\big(\prod_{i=1}^{k}\lambda_i(\exp(\theta Y))\big) \le \frac{1}{\theta}\mathbb{E}\log\mathrm{Tr}_k\big[\exp(\theta Y)\big].\end{aligned}$$

We have used the fact that $\prod_{i=1}^{k}\lambda_i(A) \le \mathrm{Tr}_k[A]$ for any $A \in \mathbf{H}_n^+$. Next we define the random matrices $A^{(i)} = \exp(\theta X^{(i)}) \in \mathbf{H}_n^{++}, 1 \le i \le m$. Since $X^{(i)}, 1 \le i \le m$, are independent, $A^{(i)}, 1 \le i \le m$, are also independent. Therefore we may apply inequality (5.38b) in Lemma 5.5.1 to get

$$\begin{aligned}\mathbb{E}\log\mathrm{Tr}_k\big[\exp(\theta Y)\big] &= \mathbb{E}\log\mathrm{Tr}_k\big[\exp\big(\sum_{i=1}^{m}\theta X^{(i)}\big)\big]\\ &= \mathbb{E}\log\mathrm{Tr}_k\big[\exp\big(\sum_{i=1}^{m}\log A^{(i)}\big)\big]\\ &\le \log\mathrm{Tr}_k\big[\exp\big(\sum_{i=1}^{m}\log\mathbb{E}A^{(i)}\big)\big]\\ &= \log\mathrm{Tr}_k\big[\exp\big(\sum_{i=1}^{m}\log\mathbb{E}\exp(\theta X^{(i)})\big)\big].\end{aligned}$$

Since $\theta > 0$ is arbitrary, we thus have

$$\mathbb{E}\sum_{i=1}^{k}\lambda_i(Y) \le \inf_{\theta>0}\ \frac{1}{\theta}\log\mathrm{Tr}_k\big[\exp\big(\sum_{i=1}^{m}\log\mathbb{E}\exp(\theta X^{(i)})\big)\big].$$

The proof of (5.5a) shares a similar spirit, except that we use (5.38a) instead of (5.38b). For any $t \in \mathbb{R}, \theta > 0$, we use the Markov's inequality to obtain

$$\begin{aligned}\mathbb{P}\left\{\sum_{i=1}^{k}\lambda_i(Y) \ge t\right\} &= \mathbb{P}\left\{\exp\Big(\frac{\theta}{k}\sum_{i=1}^{k}\lambda_i(Y)\Big) \ge e^{\frac{\theta t}{k}}\right\}\\ &\le e^{-\frac{\theta t}{k}}\mathbb{E}\exp\Big(\frac{1}{k}\sum_{i=1}^{k}\lambda_i(\theta Y)\Big) \le e^{-\frac{\theta t}{k}}\mathbb{E}\Big[\big(\mathrm{Tr}_k\exp(\theta Y)\big)^{\frac{1}{k}}\Big].\end{aligned}$$

Then again by defining $A^{(i)} = \exp(\theta X^{(i)}) \in \mathbf{H}_n^{++}, 1 \le i \le m$, we may apply inequality (5.38a) in Lemma 5.5.1 to obtain

$$\mathbb{E}\Big[\big(\mathrm{Tr}_k\exp(\theta Y)\big)^{\frac{1}{k}}\Big] \le \Big(\mathrm{Tr}_k\exp\big(\sum_{i=1}^{m}\log\mathbb{E}\exp(\theta X^{(i)})\big)\Big)^{\frac{1}{k}}$$

Since $\theta > 0$ is arbitrary, we thus have

$$\mathbb{P}\left\{\sum_{i=1}^{k}\lambda_i(Y) \geq t\right\} \leq \inf_{\theta>0} e^{-\frac{\theta t}{k}}\left(\mathrm{Tr}_k \exp\left(\sum_{i=1}^{m}\log \mathbb{E}\exp(\theta X^{(i)})\right)\right)^{\frac{1}{k}}.$$

We proceed to (5.4b) and (5.5b). The first inequality (5.4b) can be similarly verified by noticing that the sum of a Hermitian matrix's $k$ smallest eigenvalues is a concave function of the matrix itself. For the second inequality in (5.4b) and inequality (5.5b), we only need to consider arbitrary $\theta < 0$, and use the fact that $\theta\lambda_{n-i+1}(A) = \lambda_i(\theta A)$ for any $A \in \mathbf{H}_n$. Then repeating the arguments for (5.4a) and (5.5a), we can similarly show that

$$\mathbb{E}\sum_{i=1}^{k}\lambda_{n-i+1}(Y) \geq \sup_{\theta<0}\frac{1}{\theta}\log \mathrm{Tr}_k\left[\exp\left(\sum_{i=1}^{m}\log \mathbb{E}\exp(\theta X^{(i)})\right)\right],$$

and

$$\mathbb{P}\left\{\sum_{i=1}^{k}\lambda_{n-i+1}(Y) \leq t\right\} \leq \inf_{\theta<0} e^{-\frac{\theta t}{k}}\left(\mathrm{Tr}_k \exp\left(\sum_{i=1}^{m}\log \mathbb{E}\exp(\theta X^{(i)})\right)\right)^{\frac{1}{k}}.$$

□

With all the preceding results, we are now ready to prove our Chernoff-type bounds on the sum of the $k$ largest (or smallest) eigenvalues. The following proof is an imitation of Tropp's proof of Theorem 5.1.1 in [122].

***Proof of Theorem 5.1.4***. Since $0 \leq \lambda_n(X^{(i)}) \leq \lambda_1(X^{(i)}) \leq c$, we can use lemma 5.4.1 in [122] to obtain the estimate

$$\log \mathbb{E}\exp(\theta X^{(i)}) \leq \frac{e^{\theta c}-1}{c}\mathbb{E}X^{(i)} = g(\theta)\mathbb{E}X^{(i)}, \quad \theta \in \mathbb{R},$$

where $g(\theta) = \frac{e^{\theta c}-1}{c}$. So we have the following

$$\begin{aligned}
\mathrm{Tr}_k\left[\exp\left(\sum_{i=1}^{m}\log \mathbb{E}\exp(\theta X^{(i)})\right)\right] &\leq \mathrm{Tr}_k\left[\exp\left(g(\theta)\sum_{i=1}^{m}\mathbb{E}X^{(i)}\right)\right] \\
&= \mathrm{Tr}_k\left[\exp\left(g(\theta)\mathbb{E}Y\right)\right] \\
&\leq \binom{n}{k}\prod_{i=1}^{k}\lambda_i\left(\exp\left(g(\theta)\mathbb{E}Y\right)\right) \\
&= \binom{n}{k}\exp\left(\sum_{i=1}^{k}\lambda_i(g(\theta)\mathbb{E}Y)\right).
\end{aligned} \tag{5.39}$$

The second inequality above is due to the fact that $\mathrm{Tr}_k[A] \leq \binom{n}{k} \prod_{i=1}^{k} \lambda_i(A)$ for any $A \in \mathbf{H}_n^+$. Notice that for $\theta > 0$, $g(\theta) = \frac{e^{\theta c}-1}{c} > 0$. We then apply (5.39) to (5.4a) in Theorem 5.1.2 to get

$$\begin{aligned}\mathbb{E}\sum_{i=1}^{k}\lambda_i(Y) &\leq \inf_{\theta>0}\frac{1}{\theta}\log\left(\binom{n}{k}\exp\big(\sum_{i=1}^{k}\lambda_i(g(\theta)\mathbb{E}Y)\big)\right)\\ &= \inf_{\theta>0}\frac{g(\theta)}{\theta}\sum_{i=1}^{k}\lambda_i(\mathbb{E}Y)+\frac{1}{\theta}\log\binom{n}{k}.\end{aligned}$$

As mentioned in [122], this infimum does not admit a closed form. By making change of variable $\theta \to \theta/c$, we obtain (5.8a)

$$\mathbb{E}\sum_{i=1}^{k}\lambda_i(Y) \leq \inf_{\theta>0}\frac{e^{\theta}-1}{\theta}\sum_{i=1}^{k}\lambda_i(\mathbb{E}Y)+\frac{c}{\theta}\log\binom{n}{k}.$$

Similarly, we apply (5.39) to (5.5a) in Theorem 5.1.2 to get

$$\begin{aligned}\mathbb{P}\left\{\sum_{i=1}^{k}\lambda_i(Y)\geq t\right\} &\leq \inf_{\theta>0} e^{-\frac{\theta t}{k}}\left(\binom{n}{k}\exp\big(\sum_{i=1}^{k}\lambda_i(g(\theta)\mathbb{E}Y)\big)\right)^{\frac{1}{k}}\\ &= \inf_{\theta>0} e^{-\frac{\theta t}{k}}\binom{n}{k}^{\frac{1}{k}}\exp\Big(\frac{g(\theta)}{k}\sum_{i=1}^{k}\lambda_i(\mathbb{E}Y)\Big).\end{aligned}$$

If we choose $t = (1+\varepsilon)\sum_{i=1}^{k}\lambda_i(\mathbb{E}Y)$ for $\varepsilon \geq 0$, we have

$$\begin{aligned}&\mathbb{P}\left\{\sum_{i=1}^{k}\lambda_i(Y)\geq(1+\varepsilon)\sum_{i=1}^{k}\lambda_i(\mathbb{E}Y)\right\}\\ &\qquad\leq \inf_{\theta>0}\binom{n}{k}^{\frac{1}{k}}\exp\Big((g(\theta)-(1+\varepsilon)\theta)\frac{1}{k}\sum_{i=1}^{k}\lambda_i(\mathbb{E}Y)\Big).\end{aligned}$$

Minimizing the right hand side with $\theta = \frac{\log(1+\varepsilon)}{c}$ gives (5.9a).

Now consider $\theta < 0$, we have $g(\theta) = \frac{e^{\theta c}-1}{c} < 0$. We then apply (5.39) to (5.4b) in Theorem 5.1.2 to get

$$\begin{aligned}\mathbb{E}\sum_{i=1}^{k}\lambda_{n-i+1}(Y) &\geq \sup_{\theta<0}\frac{1}{\theta}\log\left(\binom{n}{k}\exp\big(\sum_{i=1}^{k}\lambda_i(g(\theta)\mathbb{E}Y)\big)\right)\\ &= \sup_{\theta<0}\frac{g(\theta)}{\theta}\sum_{i=1}^{k}\lambda_{n-i+1}(\mathbb{E}Y)+\frac{1}{\theta}\log\binom{n}{k}.\end{aligned}$$

We have used $\lambda_i(g(\theta)\mathbb{E}Y) = g(\theta)\lambda_{n-i+1}(\mathbb{E}Y)$ when $g(\theta) < 0$. By making change of variable $\theta \to -\theta/c$, we obtain (5.8b)

$$\mathbb{E}\sum_{i=1}^{k}\lambda_{n-i+1}(Y) \geq \sup_{\theta>0}\frac{1-e^{-\theta}}{\theta}\sum_{i=1}^{k}\lambda_{n-i+1}(\mathbb{E}Y) - \frac{c}{\theta}\log\binom{n}{k}.$$

Again, we apply (5.39) to (5.5b) in Theorem 5.1.2 to get

$$\begin{aligned}\mathbb{P}\left\{\sum_{i=1}^{k}\lambda_{n-i+1}(Y) \leq t\right\} &\leq \inf_{\theta<0} e^{-\frac{\theta t}{k}}\left(\binom{n}{k}\exp\big(\sum_{i=1}^{k}\lambda_i(g(\theta)\mathbb{E}Y)\big)\right)^{\frac{1}{k}} \\ &= \inf_{\theta<0} e^{-\frac{\theta t}{k}}\binom{n}{k}^{\frac{1}{k}}\exp\Big(\frac{g(\theta)}{k}\sum_{i=1}^{k}\lambda_{n-i+1}(\mathbb{E}Y)\Big).\end{aligned}$$

If we choose $t = (1-\varepsilon)\sum_{i=1}^{k}\lambda_{n-i+1}(\mathbb{E}Y)$ for $\varepsilon \in [0,1)$, we have

$$\begin{aligned}&\mathbb{P}\left\{\sum_{i=1}^{k}\lambda_{n-i+1}(Y) \leq (1-\varepsilon)\sum_{i=1}^{k}\lambda_{n-i+1}(\mathbb{E}Y)\right\} \\ &\qquad \leq \inf_{\theta<0}\binom{n}{k}^{\frac{1}{k}}\exp\Big(\big(g(\theta)-(1-\varepsilon)\theta\big)\frac{1}{k}\sum_{i=1}^{k}\lambda_{n-i+1}(\mathbb{E}Y)\Big).\end{aligned}$$

Minimizing the right hand sides with $\theta = \frac{\log(1-\varepsilon)}{c}$ gives (5.9b). □

## 5.6 Supporting Materials

### 5.6.1 Mixed Discriminant

The mixed discriminant $D(A^{(1)}, A^{(2)}, \cdots, A^{(n)})$ of $n$ matrices $A^{(1)}, A^{(2)}, \cdots, A^{(n)} \in \mathbb{C}^{n\times n}$ is defined as

$$D(A^{(1)}, A^{(2)}, \cdots, A^{(n)}) = \frac{1}{n!}\sum_{\sigma\in S_n}\det\begin{bmatrix} A_{11}^{(\sigma(1))} & A_{12}^{(\sigma(2))} & \cdots & A_{1n}^{(\sigma(n))} \\ A_{21}^{(\sigma(1))} & A_{22}^{(\sigma(2))} & \cdots & A_{1n}^{(\sigma(n))} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1}^{(\sigma(1))} & A_{n2}^{(\sigma(2))} & \cdots & A_{nn}^{(\sigma(n))} \end{bmatrix}, \quad (5.40)$$

where $S_n$ denotes the symmetric group of order $n$. We here list some basic facts about mixed discriminants. For more properties of mixed discriminants, one may refer to [6, 91].

- Symmetry: $D(A^{(1)}, A^{(2)}, \cdots, A^{(n)})$ is symmetric in $A^{(1)}, A^{(2)}, \cdots, A^{(n)}$, i.e.

$$D(A^{(1)}, A^{(2)}, \cdots, A^{(n)}) = D(A^{\sigma(1)}, A^{\sigma(2)}, \cdots, A^{\sigma(n)}), \quad \sigma \in S_n.$$

- Multilinearity: for any $\alpha, \beta \in \mathbb{R}$,

$$D(\alpha A+\beta B, A^{(2)}, \cdots, A^{(n)}) = \alpha D(A, A^{(2)}, \cdots, A^{(n)})+\beta D(B, A^{(2)}, \cdots, A^{(n)}).$$

- Positiveness [6]: If $A^{(1)}, A^{(2)}, \cdots, A^{(n)} \in \mathbf{H}_n^+$, then $D(A^{(1)}, A^{(2)}, \cdots, A^{(n)}) \geq 0$; if $A^{(1)}, A^{(2)}, \cdots, A^{(n)} \in \mathbf{H}_n^{++}$, then $D(A^{(1)}, A^{(2)}, \cdots, A^{(n)}) > 0$.

The relation between the mixed discriminant and $\mathrm{Tr}_k$ is obvious. If we calculate the mixed discriminant for $k$ copies of $A \in \mathbb{C}^{n\times n}$ and $n-k$ copies of $I_n$, we can find that

$$D(\underbrace{A, \cdots, A}_{k}, \underbrace{I_n, \cdots, I_n}_{n-k}) = \binom{n}{k}^{-1} \sum_{1\leq i_1<i_2<\cdots<i_k\leq n} \det[A_{(i_1\cdots i_k, i_1\cdots i_k)}] = \binom{n}{k}^{-1} \mathrm{Tr}_k[A]. \tag{5.41}$$

This is why the mixed discriminant plays an important role in the proof of our main theorem. In particular, we will need the following inequality on mixed discriminant by Alexandrov [2].

**Theorem 5.6.1** (Alexandrov–Fenchel Inequality for Mixed Discriminants)**.** *For any $B \in \mathbf{H}_n$ and any $A, \underbrace{A^{(3)}, \cdots, A^{(n)}}_{n-2} \in \mathbf{H}_n^{++}$, we have*

$$D(A, B, A^{(3)}, \cdots, A^{(n)})^2 \geq D(A, A, A^{(3)}, \cdots, A^{(n)})D(B, B, A^{(3)}, \cdots, A^{(n)}), \tag{5.42}$$

*with equality if and only if $B = \lambda A$ for some $\lambda \in \mathbb{R}$.*

This theorem originally applied to real symmetric matrices when established. A proof of its extension to Hermitian matrices can be found in [66]. By continuity, inequality (5.42) can extend to the case that $A, A^{(3)}, \cdots, A^{(n)} \in \mathbf{H}_n^+$, but the necessity of the condition for equality is no longer valid.

Repeatedly applying the Alexandrov–Fenchel inequality (5.42) grants us the following corollary.

**Corollary 5.6.2.** *For any $0 \leq l \leq k \leq n$, and any $A, B, \underbrace{A^{(k+1)}, \cdots, A^{(n)}}_{n-k} \in \mathbf{H}_n^+$, we have*

$$\begin{aligned} &D(\underbrace{A, \cdots, A}_{l}, \underbrace{B, \cdots, B}_{k-l}, \underbrace{A^{(k+1)}, \cdots, A^{(n)}}_{n-k})^k \\ \geq\ &D(\underbrace{A, \cdots, A}_{k}, \underbrace{A^{(k+1)}, \cdots, A^{(n)}}_{n-k})^l \cdot D(\underbrace{B, \cdots, B}_{k}, \underbrace{A^{(k+1)}, \cdots, A^{(n)}}_{n-k})^{k-l}. \end{aligned} \tag{5.43}$$

A direct result of Corollary 5.6.2 is the following general Brunn–Minkowski theorem for mixed discriminants.

**Corollary 5.6.3.** ***(General Brunn–Minkowski Theorem for Mixed Discriminants)*** *For any $1 \leq k \leq n$, and any fixed $\underbrace{A^{(k+1)}, \cdots, A^{(n)}}_{n-k} \in \mathbf{H}_n^+$, the function*

$$\begin{aligned} \mathbf{H}_n^+ &\longrightarrow \mathbb{R} \\ A &\longmapsto D(\underbrace{A, \cdots, A}_{k}, \underbrace{A^{(k+1)}, \cdots, A^{(n)}}_{n-k})^{\frac{1}{k}} \end{aligned} \tag{5.44}$$

*is concave.*

*Proof.* Fixing $A^{(k+1)}, \cdots, A^{(n)}$, we will use $D(A[k])$ and $D(A[l], B[k-l])$ to denote

$$D(\underbrace{A, \cdots, A}_{k}, \underbrace{A^{(k+1)}, \cdots, A^{(n)}}_{n-k})^{\frac{1}{k}} \text{ and } D(\underbrace{A, \cdots, A}_{l}, \underbrace{B, \cdots, B}_{k-l}, \underbrace{A^{(k+1)}, \cdots, A^{(n)}}_{n-k})$$

respectively. For any $A, B \in \mathbf{H}_n^+$, and any $\tau \in [0, 1]$, using the multilinearity of mixed discriminants and Corollary 5.6.2, we have

$$\begin{aligned} D((\tau A + (1-\tau)B)[k]) &= \sum_{l=0}^{k} \binom{k}{l} \tau^l (1-\tau)^{k-l} D(A[l], B[k-l]) \\ &\geq \sum_{l=0}^{k} \binom{k}{l} \tau^l (1-\tau)^{k-l} D(A[k])^{\frac{l}{k}} D(B[k])^{\frac{k-l}{k}} \\ &= (\tau D(A[k])^{\frac{1}{k}} + (1-\tau) D(B[k])^{\frac{1}{k}})^k, \end{aligned}$$

that is $D((\tau A + (1-\tau)B)[k])^{\frac{1}{k}} \geq \tau D(A[k])^{\frac{1}{k}} + (1-\tau) D(B[k])^{\frac{1}{k}}$. □

If we choose $A^{(k+1)}, \cdots, A^{(n)}$ to be $n-k$ copies of $I_n$, Corollary 5.6.3 immediately implies that the function $A \mapsto (\mathrm{Tr}_k[A])^{\frac{1}{k}}$ is concave on $\mathbf{H}_n^+$, which is a special case of Theorem 5.3.1 with $H = \mathbf{0}$. So we see the connection between the Alexandrov–Fenchel inequality and our generalized Lieb's theorem. However, the arguments in the proof of Corollary 5.6.3 do not seem to work with $H \neq \mathbf{0}$. We hence need more tools to handle the more general case.

### 5.6.2 Exterior Algebra

Here we give a brief review of exterior algebras on the vector space $\mathbb{C}^n$. For more details, one may refer to [14, 100]. For the convenience of our use, the notations in our paper might be different from those in other materials. For any $1 \leq k \leq n$, let

$\wedge^k(\mathbb{C}^n)$ denote the vector space of the $k_{th}$ exterior algebra of $\mathbb{C}^n$, equipped with the inner product

$$\begin{aligned}
\langle\cdot,\cdot\rangle_{\wedge^k}: \quad & \wedge^k(\mathbb{C}^n)\times\wedge^k(\mathbb{C}^n) \longrightarrow \mathbb{C} \\
\langle u_1\wedge\cdots\wedge u_k, v_1\wedge\cdots\wedge v_k\rangle_{\wedge^k} &= \det\begin{bmatrix} \langle u_1,v_1\rangle & \langle u_1,v_2\rangle & \cdots & \langle u_1,v_k\rangle \\ \langle u_2,v_1\rangle & \langle u_2,v_2\rangle & \cdots & \langle u_2,v_k\rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle u_k,v_1\rangle & \langle u_k,v_2\rangle & \cdots & \langle u_k,v_k\rangle \end{bmatrix},
\end{aligned}$$

where $\langle u,v\rangle = u^*v$ is the standard $l_2$ inner product on $\mathbb{C}^n$.

Let $\mathcal{L}(\wedge^k(\mathbb{C}^n))$ denote the space of all linear operators from $\wedge^k(\mathbb{C}^n)$ to itself. For any matrices $A^{(1)}, A^{(2)}, \cdots, A^{(k)} \in \mathbb{C}^{n\times n}$, we can define an element in $\mathcal{L}(\wedge^k(\mathbb{C}^n))$:

$$\begin{aligned}
&\mathcal{M}^{(k)}(A^{(1)}, A^{(2)}, \cdots, A^{(k)}): \\
&\qquad\qquad \wedge^k(\mathbb{C}^n) \longrightarrow \wedge^k(\mathbb{C}^n) \\
&\qquad v_1\wedge v_2\wedge\cdots\wedge v_k \longmapsto \sum_{\sigma\in S_k} A^{(\sigma(1))}v_1\wedge A^{(\sigma(2))}v_2\wedge\cdots\wedge A^{(\sigma(k))}v_k,
\end{aligned} \tag{5.45}$$

where $S_k$ is the symmetric group of order $k$. Apparently, the map

$$(A^{(1)}, A^{(2)}, \cdots, A^{(k)}) \longmapsto \mathcal{M}^{(k)}(A^{(1)}, A^{(2)}, \cdots, A^{(k)})$$

is symmetric in $A^{(1)}, A^{(2)}, \cdots, A^{(k)}$ and is linear in each single $A^{(i)}$. For simplicity, we will use the following notations for any matrices $A, B, C \in \mathbb{C}^{n\times n}$:

$$\mathcal{M}_0^{(k)}(A) = \frac{1}{k!}\mathcal{M}^{(k)}(A,\cdots,A), \tag{5.46a}$$

$$\mathcal{M}_1^{(k)}(A;B) = \frac{1}{(k-1)!}\mathcal{M}^{(k)}(A,B,\cdots,B), \tag{5.46b}$$

$$\mathcal{M}_2^{(k)}(A,B;C) = \frac{1}{(k-2)!}\mathcal{M}^{(k)}(A,B,C,\cdots,C). \tag{5.46c}$$

To avoid confusion, we define $\mathcal{M}_1^{(1)}(A;B) = \mathcal{M}_0^1(A)$, $\mathcal{M}_2^{(1)}(A,B;C) = \mathbf{0}$, and $\mathcal{M}_2^{(2)}(A,B;C) = \mathcal{M}_1^{(2)}(A;B)$. Obviously the identity operator in $\mathcal{L}(\wedge^k(\mathbb{C}^n))$ is $\mathcal{M}_0(I_n)$. We will be using the following properties:

- Invertibility: if $A\in\mathbb{C}^{n\times n}$ is invertible, then $(\mathcal{M}_0^{(k)}(A))^{-1} = \mathcal{M}_0^{(k)}(A^{-1})$.
- Adjoint: for any $A\in\mathbb{C}^{n\times n}$, $(\mathcal{M}_0^{(k)}(A))^* = \mathcal{M}_0^{(k)}(A^*)$, with respect to the inner product $\langle\cdot,\cdot\rangle_{\wedge^k}$.

- Positiveness: If $A \in \mathbf{H}_n$, then $\mathcal{M}_0^{(k)}(A)$ is Hermitian; if $A \in \mathbf{H}_n^+$, then $\mathcal{M}_0^{(k)}(A) \succeq \mathbf{0}$; if $A \in \mathbf{H}_n^{++}$, then $\mathcal{M}_0^{(k)}(A) \succ \mathbf{0}$.

- Product properties: for any $A, B, C, D \in \mathbb{C}^{n\times n}$, we have

$$\mathcal{M}_0^{(k)}(AB) = \mathcal{M}_0^{(k)}(A)\mathcal{M}_0^{(k)}(B), \tag{5.47a}$$
$$\mathcal{M}_1^{(k)}(A;B)\mathcal{M}_0^{(k)}(C) = \mathcal{M}_1^{(k)}(AC;BC), \tag{5.47b}$$
$$\mathcal{M}_0^{(k)}(C)\mathcal{M}_1^{(k)}(A;B) = \mathcal{M}_1^{(k)}(CA;CB), \tag{5.47c}$$
$$\mathcal{M}_1^{(k)}(A;C)\mathcal{M}_1^{(k)}(B;D) = \mathcal{M}_2^{(k)}(AD,CB;CD) + \mathcal{M}_1^{(k)}(AB;CD). \tag{5.47d}$$

- Derivative properties: for any differentiable functions $A(t), B(t) : \mathbb{R} \longrightarrow \mathbb{C}^{n\times n}$, we have

$$\frac{\partial}{\partial t}\mathcal{M}_0^{(k)}(A(t)) = \mathcal{M}_1^{(k)}(A'(t);A(t)) \tag{5.48a}$$
$$\frac{\partial}{\partial t}\mathcal{M}_1^{(k)}(A(t);B(t)) = \mathcal{M}_1^{(k)}(A'(t);B(t)) + \mathcal{M}_2^{(k)}(A(t),B'(t);B(t)). \tag{5.48b}$$

Next we consider the natural basis of $\wedge^k(\mathbb{C}^n)$,

$$\{e_{i_1} \wedge e_{i_2} \wedge \cdots \wedge e_{i_k}\}_{1\le i_1<i_2<\cdots<i_k\le n},$$

which is orthogonal under the inner product $\langle\cdot,\cdot\rangle_{\wedge^k}$. Then the trace function on $\mathcal{L}(\wedge^k(\mathbb{C}^n))$ is defined as

$$\begin{aligned}\mathrm{Tr}: \quad & \mathcal{L}(\wedge^k(\mathbb{C}^n)) \longrightarrow \mathbb{C} \\ & \mathrm{Tr}[\mathcal{F}] = \sum_{1\le i_1<i_2<\cdots<i_k\le n} \langle e_{i_1} \wedge e_{i_2} \wedge \cdots \wedge e_{i_k}, \mathcal{F}(e_{i_1} \wedge e_{i_2} \wedge \cdots \wedge e_{i_k})\rangle_{\wedge^k}.\end{aligned} \tag{5.49}$$

It is not hard to check that this trace function is also invariant under cyclic permutation, i.e. $\mathrm{Tr}[\mathcal{F}\mathcal{G}] = \mathrm{Tr}[\mathcal{G}\mathcal{F}]$ for any $\mathcal{F}, \mathcal{G} \in \mathcal{L}(\wedge^k(\mathbb{C}^n))$. Then for any $A^{(1)}, \cdots, A^{(k)} \in \mathbb{C}^{n\times n}$, the trace $\mathrm{Tr}[\mathcal{M}^{(k)}(A^{(1)}, \cdots, A^{(k)})]$ coincides with the definition of the mixed discriminant, as one can check that

$$\begin{aligned}&\mathrm{Tr}[\mathcal{M}^{(k)}(A^{(1)}, \cdots, A^{(k)})] \\ =& \sum_{\sigma\in S_k} \sum_{1\le i_1<\cdots<i_k\le n} \langle e_{i_1} \wedge \cdots \wedge e_{i_k}, A^{(\sigma(1))}e_{i_1} \wedge \cdots \wedge A^{(\sigma(k))}e_{i_k}\rangle_{\wedge^k} \\ =& \frac{n!}{(n-k)!} D(A^{(1)}, \cdots, A^{(k)}, \underbrace{I_n, \cdots, I_n}_{n-k}).\end{aligned} \tag{5.50}$$

From this observation, we can now express the $k$-trace of a matrix $A \in \mathbb{C}^{n\times n}$ as

$$\mathrm{Tr}_k[A] = \mathrm{Tr}\big[\mathcal{M}_0^{(k)}(A)\big]. \tag{5.51}$$

For those who are familiar with exterior algebra, it is clear that the spectrum of $\mathcal{M}_0^{(k)}$ is just $\{\lambda_{i_1}\lambda_{i_2}\cdots\lambda_{i_k}\}_{1\le i_1<i_2<\cdots<i_k\le n}$, where $\lambda_1, \lambda_2, \cdots, \lambda_n$ are the eigenvalues of $A$. So in this way it is more convenient to see that $\mathrm{Tr}\big[\mathcal{M}_0^{(k)}(A)\big]$ = sum(spectrum of $\mathcal{M}_0^{(k)}(A)$) = $\sum_{1\le i_1<\cdots<i_k\le n}\lambda_{i_1}\lambda_{i_2}\cdots\lambda_{i_k} = \mathrm{Tr}_k[A]$. Our proof of Theorem 5.3.1 will base on the expression (5.51).

In fact, our proof the main theorem can be done without introducing the exterior algebra. We can instead go through the whole proof only using notations of mixed discriminant. The advantage of using exterior algebra is that it interprets the $k$-trace as the normal trace of operators in a space of higher dimension, so our $k$-trace functions have a nicer form that imitates the trace function in the original Lieb's theorem. Also for the same reason, we are able to construct our proof by following the arguments of Lieb's original proof in [67].

We next introduce some notations to simplify the expressions in what follows. For any $n$ real numbers $\lambda_1, \lambda_2, \cdots, \lambda_n \in \mathbb{R}$, we define the three symmetric forms

$$p^{(n,k)} = \sum_{1\le i_1<i_2<\cdots<i_k\le n} \lambda_{i_1}\lambda_{i_2}\cdots\lambda_{i_k}, \quad 1 \le k \le n, \tag{5.52a}$$

$$d_i^{(n,k)} = \sum_{\substack{1\le j_1<j_2<\cdots<j_{k-1}\le n\\ i\notin\{j_1,j_2,\cdots,j_{k-1}\}}} \lambda_{j_1}\lambda_{j_2}\cdots\lambda_{j_{k-1}}, \quad 2 \le k \le n,\ 1 \le i \le n, \tag{5.52b}$$

$$g_{ij}^{(n,k)} = \sum_{\substack{1\le l_1<l_2<\cdots<l_{k-2}\le n\\ i,j\notin\{l_1,l_2,\cdots,l_{k-2}\}}} \lambda_{l_1}\lambda_{l_2}\cdots\lambda_{l_{k-2}}, \quad 3 \le k \le n,\ 1 \le i,j \le n,\ i \ne j. \tag{5.52c}$$

For consistency, we define $d_i^{(n,k)} = 1$ if $k = 1$; $g_{ij}^{(n,k)} = 1$ if $k = 2$ and $i \ne j$; $g_{ij}^{(n,k)} = 0$ if $k = 1$ or $i = j$. Also we define $p^{(n,k)} = d_i^{(n,k)} = g_{ij}^{(n,k)} = 0$ if $k > n$. Throughout this paper, whenever we are given some real numbers $\lambda_1, \lambda_2, \cdots, \lambda_n$, the quantities $p^{(n,k)}, d_i^{(n,k)}, g_{ij}^{(n,k)}$ are always defined correspondingly with respect to $\{\lambda_i\}_{1\le i\le n}$. The following relations are easy to verify with the definitions above, and will be useful in our proofs of lemmas and theorems. For any $n, k$, and any $1 \le i, j \le n$ such that $i \ne j$, we have the expansion relations

$$p^{(n,k)} = \lambda_i d_i^{(n,k)} + d_i^{(n,k+1)}, \qquad d_i^{(n,k)} = \lambda_j g_{ij}^{(n,k)} + g_{ij}^{(n,k+1)}. \tag{5.53}$$

With the notations defined above, we give the following lemma. The proof is straightforward by definition, so we omit it here.

**Lemma 5.6.4.** *For any $A, B \in \mathbb{C}^{n\times n}$, and any diagonal matrix $\Lambda \in \mathbb{C}^{n\times n}$ with diagonal entries $\lambda_1, \lambda_2, \cdots, \lambda_n$, we have the following identities*

$$\mathrm{Tr}[\mathcal{M}_0^{(k)}(\Lambda)] = p^{(n,k)}, \tag{5.54a}$$

$$\mathrm{Tr}[\mathcal{M}_1^{(k)}(A;\Lambda)] = \sum_{i=1}^{n} A_{ii} d_i^{(n,k)}, \tag{5.54b}$$

$$\mathrm{Tr}[\mathcal{M}_2^{(k)}(A,B;\Lambda)] = \sum_{1\le i,j\le n} (A_{ii}B_{jj} - A_{ji}B_{ij}) g_{ij}^{(n,k)}, \tag{5.54c}$$

*for all $1 \le k \le n$, where $p^{(n,k)}, d_i^{(n,k)}, g_{ij}^{(n,k)}$ are defined with respect to $\lambda_1, \lambda_2, \cdots, \lambda_n$.*

We here provide an alternative of Lemma 5.4.4 using the following lemma.

**Lemma 5.6.5.** *For any $A \in \mathbf{H}_n$, we have*

$$\mathcal{M}_0^{(k)}(\exp(A)) = \exp\left(\mathcal{M}_1^{(k)}(A; I_n)\right).$$

*Proof.* We need to show that for any $v_1 \wedge v_2 \wedge \cdots \wedge v_k \in \wedge^k(\mathbb{C}^n)$,

$$\mathcal{M}_0^{(k)}(\exp(A))(v_1 \wedge v_2 \wedge \cdots \wedge v_k) = \exp\left(\mathcal{M}_1^{(k)}(A; I_n)\right)(v_1 \wedge v_2 \wedge \cdots \wedge v_k). \tag{5.55}$$

We use Taylor expansion of $e^x$ to expand

$$\mathcal{M}_0^{(k)}(\exp(A)) = \mathcal{M}_0^{(k)}\Big(\sum_{j=0}^{+\infty} \frac{1}{j!} A^j\Big), \quad \exp\left(\mathcal{M}_1^{(k)}(A; I_n)\right) = \sum_{j=0}^{+\infty} \frac{1}{j!} \left(\mathcal{M}_1^{(k)}(A; I_n)\right)^j.$$

Then for any integers $j_1, j_2, \ldots, j_k \ge 0$, the coefficient of the term $A^{j_1}v_1 \wedge A^{j_2}v_2 \wedge \cdots \wedge A^{j_k}v_k$ in the left hand side of (5.55) is

$$\frac{1}{j_1! j_2! \cdots j_k!},$$

and the coefficient of the same term in the right hand side of (5.55) is also

$$\frac{1}{J!}\binom{J}{j_1}\binom{J-j_1}{j_2}\cdots\binom{J-j_1-j_2-\cdots-j_{k-1}}{j_k} = \frac{1}{j_1! j_2! \cdots j_k!},$$

where $(J = j_1 + j_2 + \cdots + j_k)$. □

***An alternative proof of Lemma 5.4.4***. Using Lemma 5.6.5 and the original GT inequality for normal trace, we have

$$\begin{aligned}
\mathrm{Tr}_k[\exp(A+B)] &= \mathrm{Tr}[\mathcal{M}_0^{(k)}(\exp(A+B))] \\
&= \mathrm{Tr}[\exp(\mathcal{M}_1^{(k)}(A+B;I_n))] \\
&= \mathrm{Tr}[\exp(\mathcal{M}_1^{(k)}(A;I_n)+\mathcal{M}_1^{(k)}(B;I_n))] \\
&\leq \mathrm{Tr}[\exp(\mathcal{M}_1^{(k)}(A;I_n))\exp(\mathcal{M}_1^{(k)}(B;I_n))] \\
&= \mathrm{Tr}[\mathcal{M}_0^{(k)}(\exp(A))\mathcal{M}_0^{(k)}(\exp(B))] \\
&= \mathrm{Tr}_k[\exp(A)\exp(B)],
\end{aligned}$$

where we have used that $\mathcal{M}_1^{(k)}(X;I_n)$ is linear in $X$. As shown by Petz [93], in the original GT inequality, the equality $\mathrm{Tr}[\exp(A+B)] = \mathrm{Tr}[\exp(A)\exp(B)]$ holds for $A, B \in \mathbf{H}_n$ if and only if $AB = BA$. Therefore, according to our calculation above, the equality $\mathrm{Tr}_k[\exp(A+B)] = \mathrm{Tr}_k[\exp(A)\exp(B)]$ holds if and only if

$$\mathcal{M}_1^{(k)}(A;I_n)\mathcal{M}_1^{(k)}(B;I_n) = \mathcal{M}_1^{(k)}(B;I_n)\mathcal{M}_1^{(k)}(A;I_n). \tag{5.56}$$

However, one can check by definition that (5.56) is true if and only if $AB = BA$. □

### 5.6.3 Derivatives of Some Matrix Functions

Let us remind ourselves that a basic but important way to prove concavity of a differentiable function $f(t)$ is by showing that $f''(t) \leq 0$. Similarly, one way to prove concavity of a differentiable multivariate function $f(\mathbf{x})$ is by showing that the second directional derivative $\frac{\partial^2}{\partial t^2} f(\mathbf{x}+t\mathbf{y})|_{t=0} \leq 0$ for all allowed direction $\mathbf{y}$. We will use this idea to prove the concavity of the $k$-trace functions (5.18) and (5.19). For this purpose, we would need the following matrix derivative formulas.

- Consider a function $A(t) : (a,b) \longrightarrow \mathbf{H}_n$, such that $A(t)$ is differentiable on $(a,b)$, then we have[132]

$$\frac{\partial}{\partial t}\exp(A(t)) = \int_0^1 \exp(sA(t))A'(t)\exp((1-s)A(t))ds. \tag{5.57}$$

  $A'(t)$ denotes the derivative of $A(t)$ with respect to $t$.

- Consider a function $A(t) : (a,b) \longrightarrow \mathbf{H}_n^{++}$, such that $A(t)$ is differentiable on $(a,b)$, then we have[67]

$$\frac{\partial}{\partial t}(A(t))^{-1} = -(A(t))^{-1}A'(t)(A(t))^{-1}, \tag{5.58}$$

  and

$$\frac{\partial}{\partial t}\log(A(t)) = \int_0^\infty (A(t)+\tau I_n)^{-1}A'(t)(A(t)+\tau I_n)^{-1}d\tau. \tag{5.59}$$

### 5.6.4 Operator Interpolation

One of our main tools is Stein's interpolation of linear operators [116], that was developed from Hirschman's stronger version of the Hadamard three-line theorem [51]. This technique was recently adopted by Sutter et al. [119] to establish a multivariate extension of the Golden–Thompson inequality, which inspired our use of interpolation in proving the generalized Lieb's concavity theorem. We will follow the notations in [119]. For any $\theta \in (0,1)$, we define a density $\beta_\theta(t)$ on $\mathbb{R}$ by

$$\beta_\theta(t) = \frac{\sin(\pi\theta)}{2\theta(\cosh(\pi t) + \cos(\pi\theta))}, \quad t \in \mathbb{R}. \tag{5.60}$$

Specially, we define

$$\beta_0(t) = \lim_{\theta \searrow 0} \beta_\theta(t) = \frac{\pi}{2(\cosh(\pi t) + 1)}, \quad \text{and} \quad \beta_1(t) = \lim_{\theta \nearrow 1} \beta_\theta(t) = \delta(t).$$

$\beta_\theta(t)$ is a density since $\beta_\theta(t) \geq 0, t \in \mathbb{R}$ and $\int_{-\infty}^{+\infty} \beta_\theta(t)dt = 1$. We will always use $\mathcal{S}$ to denote a vertical strip on the complex plane $\mathbb{C}$:

$$\mathcal{S} = \{z \in \mathbb{C} : 0 \leq \mathrm{Re}(z) \leq 1\}. \tag{5.61}$$

The idea of complex interpolation originates from an important result in harmonic analysis, the Hadamard three-lines theorem [45], that if $f(z)$ is uniformly bounded on $\mathcal{S} = \{z \in \mathbb{C} : 0 \leq \mathrm{Re}(z) \leq 1\}$, holomorphic in the interior and continuous on the boundary, then $g(x) = \log \sup_y |f(x+iy)|$ is a convex function on $[0,1]$. Hirschman [51] improved this theorem to the following.

**Theorem 5.6.6** (Hirschman)**.** *Let $f(z)$ be uniformly bounded on $\mathcal{S}$, holomorphic in the interior and continuous on the boundary. Then for $\theta \in (0,1)$, we have*

$$\log|f(\theta)| \leq \int_{-\infty}^{+\infty} dt(\beta_{1-\theta}(t)\log|f(it)|^{1-\theta} + \beta_\theta(t)\log|f(1+it)|^\theta).$$

*Moreover, the assumption that $f(z)$ is uniformly bounded can be relaxed to*

$$\log|f(z)| \leq Ce^{a\mathrm{Im}(z)}, \quad \forall z \in \mathcal{S}, \quad \textit{for some constants } C < +\infty,\ a < \pi.$$

Stein [116] further generalized this complex interpolation theory to interpolation of linear operators.

**Theorem 5.6.7** (Stein-Hirschman)**.** *Let $G(z)$ be a map from $\mathcal{S}$ to bounded linear operators on a separable Hilbert space that is holomorphic in the interior of $\mathcal{S}$ and continuous on the boundary. Let $p_0, p_1 \in [1, +\infty], \theta \in [0,1]$, and define $p_\theta$ by*

$$\frac{1}{p_\theta} = \frac{1-\theta}{p_0} + \frac{\theta}{p_1}.$$

*Then if $\|G(z)\|_{p_{\mathrm{Re}(z)}}$ is uniformly bounded on $\mathcal{S}$, the following inequality holds:*

$$\log \|G(\theta)\|_{p_\theta} \leq \int_{-\infty}^{+\infty} dt \Big( \beta_{1-\theta}(t) \log \|G(it)\|_{p_0}^{1-\theta} + \beta_\theta(t) \log \|G(1+it)\|_{p_1}^{\theta} \Big). \quad (5.62)$$

A $k$-trace analog of the above theorem, that is used in the proof of Lemma 5.3.2, is as follows (recall that we write $\phi(\cdot) = \mathrm{Tr}_k[\cdot]^{\frac{1}{k}}$):

**Lemma 5.6.8.** *Let $G(z) : \mathcal{S} \to \mathbb{C}^{n\times n}$ be holomorphic in the interior of $\mathcal{S}$ and continuous on the boundary. Let $p_0, p_1 \in [1, +\infty]$, $\theta \in [0, 1]$, and define $p_\theta$ by*

$$\frac{1}{p_\theta} = \frac{1-\theta}{p_0} + \frac{\theta}{p_1}.$$

*Then if $\|G(z)\|$ is uniformly bounded on $\mathcal{S}$, the following inequality holds:*

$$\begin{aligned} &\log \left[ \phi(|G(\theta)|^{p_\theta})^{\frac{1}{p_\theta}} \right] \\ &\leq \int_{-\infty}^{+\infty} dt \Big( \beta_{1-\theta}(t) \log \left[ \phi(|G(it)|^{p_0})^{\frac{1-\theta}{p_0}} \right] + \beta_\theta(t) \log \left[ \phi(|G(1+it)|^{p_1})^{\frac{\theta}{p_1}} \right] \Big). \end{aligned} \quad (5.63)$$

*Proof.* For any $X \in \mathbb{C}^{n\times n}$ and $p \in [1, +\infty)$, we have that

$$\begin{aligned} \mathcal{M}_0^{(k)}(|X|^p) &= \mathcal{M}_0^{(k)}((X^*X)^{\frac{p}{2}}) = (\mathcal{M}_0^{(k)}(X^*X))^{\frac{p}{2}} \\ &= ((\mathcal{M}_0^{(k)}(X))^* \mathcal{M}_0^{(k)}(X))^{\frac{p}{2}} = |\mathcal{M}_0^{(k)}(X)|^p, \end{aligned}$$

and thus

$$\mathrm{Tr}_k\left[|X|^p\right]^{\frac{1}{p}} = \mathrm{Tr}\left[\mathcal{M}_0^{(k)}(|X|^p)\right]^{\frac{1}{p}} = \mathrm{Tr}\left[|\mathcal{M}_0^{(k)}(X)|^p\right]^{\frac{1}{p}} = \|\mathcal{M}_0^{(k)}(X)\|_p.$$

The above equality also holds for $p \to +\infty$ since we are dealing with finite-dimensional operators. If $G(z)$ is holomorphic in the interior of $\mathcal{S}$ and continuous on the boundary, then so is $\mathcal{M}_0^{(k)}(G(z))$. And if $\|G(z)\|$ is uniformly bounded on $\mathcal{S}$, then $\|\mathcal{M}_0^{(k)}(G(z))\|_{p_{\mathrm{Re}(z)}}$ is also uniformly bounded on $\mathcal{S}$, since all norms are equivalent for finite-dimensional operators. Therefore we can use Theorem 5.6.7 with $G(z)$ replaced by $\mathcal{M}_0^{(k)}(G(z))$ to get

$$\begin{aligned} &\log \left(\mathrm{Tr}_k\left[|G(\theta)|^{p_\theta}\right]^{\frac{1}{p_\theta}}\right) \\ &\leq \int_{-\infty}^{+\infty} dt \Big( \beta_{1-\theta}(t) \log \left(\mathrm{Tr}_k\left[|G(it)|^{p_0}\right]^{\frac{1-\theta}{p_0}}\right) + \beta_\theta(t) \log \left(\mathrm{Tr}_k\left[|G(1+it)|^{p_1}\right]^{\frac{\theta}{p_1}}\right) \Big). \end{aligned}$$

We then multiply both sides by $\frac{1}{k}$ to obtain (5.63). □

For $p_0, p_1 \in [1, +\infty)$, we can rewrite inequality (5.63) as

$$\begin{aligned} &\log \phi(|G(\theta)|^{p_\theta}) \\ &\leq \int_{-\infty}^{+\infty} dt\Big(\frac{(1-\theta)p_\theta}{p_0}\beta_{1-\theta}(t)\log\phi\big(|G(it)|^{p_0}\big) + \frac{\theta p_\theta}{p_1}\beta_\theta(t)\log\phi\big(|G(1+it)|^{p_1}\big)\Big) \end{aligned} \tag{5.64}$$

Notice that

$$\int_{-\infty}^{+\infty}\Big(\frac{(1-\theta)p_\theta}{p_0}\beta_{1-\theta}(t) + \frac{\theta p_\theta}{p_1}\beta_\theta(t)\Big)dt = 1.$$

Then using Jensen's inequality on the concavity of logarithm, we can immediately conclude from (5.64) that for $p_0, p_1 \in [1, +\infty)$,

$$\phi(|G(\theta)|^{p_\theta}) \leq \int_{-\infty}^{+\infty} dt\Big(\frac{(1-\theta)p_\theta}{p_0}\beta_{1-\theta}(t)\phi\big(|G(it)|^{p_0}\big) + \frac{\theta p_\theta}{p_1}\beta_\theta(t)\phi\big(|G(1+it)|^{p_1}\big)\Big), \tag{5.65}$$

under the same setting as in Lemma 5.6.8.

We remark that the operator interpolation inequality in Theorem 5.6.7 is interestingly powerful and user-friendly for proving matrix inequalities, as we have seen in the proofs of Lemma 5.3.2 and Theorem 5.3.3. In fact this technique can also prove many fundamental results in matrix theories. We here show one more example to see how we can use the trick of interpolation to prove that the map $A \mapsto A^r$ is operator concave on $\mathbf{H}_n^+$ for $r \in (0, 1]$. Note that, in general, this result is proved by using an integral expression for $A^r$ (e.g. see [24]).

We need to show that for any $A, B \in \mathbf{H}_n^+$, $\tau \in [0, 1]$,

$$\tau A^r + (1-\tau)B^r \leq C^r, \tag{5.66}$$

where $C = \tau A + (1-\tau)B$. We may assume that $C$ is invertible. The case when $C$ is not invertible can be handled by continuity. Notice that $X \leq I_n$ if and only if $\mathrm{Tr}[K^*XK] \leq \mathrm{Tr}[K^*K]$ for all $K \in \mathbb{C}^{n\times n}$. (5.66) is then equivalent to the statement that

$$\tau\mathrm{Tr}[K^*C^{-\frac{r}{2}}A^rC^{-\frac{r}{2}}K] + (1-\tau)\mathrm{Tr}[K^*C^{-\frac{r}{2}}B^rC^{-\frac{r}{2}}K] \leq \mathrm{Tr}[K^*K], \quad \forall K \in \mathbb{C}^{n\times n}.$$

Now we fix $K$ and define

$$G_X(z) = X^{\frac{z}{2}}C^{-\frac{z}{2}}K, \quad X = A, B,$$

so we have

$$\mathrm{Tr}[K^*C^{-\frac{r}{2}}X^rC^{-\frac{r}{2}}K] = \|G_X(r)\|_2^2.$$

$G_X(z)$ is holomorphic in the interior of $\mathcal{S}$ and continuous on the boundary, and $\|G_X(z)\|$ is uniformly bounded on $\mathcal{S}$. We then use inequality (5.62) in Theorem 5.6.7 with $\theta = r, p_\theta = p_0 = p_1 = 2$ to obtain

$$\|G_X(r)\|_2^2 \leq \int_{-\infty}^{+\infty} dt\Big((1-r)\beta_{1-r}(t)\|G_X(it)\|_2^2 + r\beta_r(t)\|G_X(1+it)\|_2^2\Big).$$

We have again used Jensen's inequality to get rid of the logarithms. For each $t \in \mathbb{R}$, we have that

$$\|G_X(it)\|_2^2 = \mathrm{Tr}[K^* C^{\frac{it}{2}} X^{-\frac{it}{2}} X^{\frac{it}{2}} C^{-\frac{it}{2}} K] = \mathrm{Tr}[K^* K],$$

and

$$\|G_X(1+it)\|_2^2 = \mathrm{Tr}[K^* C^{-\frac{1-it}{2}} X^{\frac{1-it}{2}} X^{\frac{1+it}{2}} C^{-\frac{1+it}{2}} K] = \mathrm{Tr}[K^* C^{-\frac{1-it}{2}} X C^{-\frac{1+it}{2}} K].$$

We then have

$$\tau\|G_A(1+it)\|_2^2 + (1-\tau)\|G_B(1+it)\|_2^2 = \mathrm{Tr}[K^* C^{-\frac{1-it}{2}} (\tau A + (1-\tau)B) C^{-\frac{1+it}{2}} K] = \mathrm{Tr}[K^* K].$$

Finally we obtain

$$\tau\|G_A(r)\|_2^2 + (1-\tau)\|G_B(r)\|_2^2 \leq \mathrm{Tr}[K^* K].$$

### 5.6.5 Homogeneous Convex/Concave Functions

**Lemma 5.6.9.** *Let $C$ be a convex cone in some linear space, i.e., $C = conv(C)$ and $C = \lambda C$ for any $\lambda > 0$. Let function $f : C \to [0, +\infty)$ be positively homogeneous of order 1, i.e., $f(\lambda x) = \lambda f(x)$, for any $x \in C$ and $\lambda > 0$. Then for any $s \in (0, 1)$, $f(x)$ is concave if and only if $f(x)^s$ is concave; for any $s \in (1, +\infty)$, $f(x)$ is convex if and only if $f(x)^s$ is convex.*

In general this lemma is proved via an argument of level sets. Here we provide a more direct proof.

*Proof.* One direction is trivial. If $f(x)$ is concave, then $f(x)^s$ is concave for $s \in (0, 1)$, since $(\cdot)^s$ is concave and monotone increasing. Conversely, if $f(x)^s$ is concave for some $s \in (0, 1)$, then $f(\tau x + (1-\tau)y)^s \geq \tau f(x)^s + (1-\tau)f(y)^s$, for any $x, y \in C, \tau \in [0, 1]$. Now given any fixed $x, y \in C, \tau \in [0, 1]$, we need to show that $\tau f(x) + (1-\tau)f(y) \leq f(\tau x + (1-\tau)y)$. If $f(x) = f(y) = 0$, then we are done.

Otherwise, we may assume that $f(x) > 0$, and define $M = \tau f(x) + (1-\tau)(f(y)+\epsilon)$ for some $\epsilon > 0$(this $\epsilon$ is not necessary if $f(y) > 0$). We then have

$$\begin{aligned} f(\tau x + (1-\tau)y) &= f\left(\frac{\tau f(x)}{M}\frac{Mx}{f(x)} + \frac{(1-\tau)(f(y)+\epsilon)}{M}\frac{My}{f(y)+\epsilon}\right)^{s\cdot\frac{1}{s}} \\ &\geq \left(\frac{\tau f(x)}{M} f\left(\frac{Mx}{f(x)}\right)^s + \frac{(1-\tau)(f(y)+\epsilon)}{M} f\left(\frac{My}{f(y)+\epsilon}\right)^s\right)^{\frac{1}{s}} \\ &= \left(\frac{\tau f(x)}{M} M^s + \frac{(1-\tau)(f(y)+\epsilon)^{1-s} f(y)^s}{M} M^s\right)^{\frac{1}{s}} \\ &= M\left(\frac{\tau f(x) + (1-\tau)(f(y)+\epsilon)^{1-s} f(y)^s}{\tau f(x) + (1-\tau)(f(y)+\epsilon)}\right)^{\frac{1}{s}}. \end{aligned}$$

We then take $\epsilon \to 0$ to obtain $f(\tau x + (1-\tau)y) \geq \tau f(x) + (1-\tau) f(y)$. Therefore $f(x)$ is concave. The convexity part can be proved similarly. □

**Lemma 5.6.10.** *Let $C$ be a convex cone in some linear space, i.e., $C = conv(C)$ and $C = \lambda C$ for any $\lambda > 0$. Let function $f : C \to [0, +\infty)$ be positively homogeneous of order 1, i.e., $f(\lambda x) = \lambda f(x)$, for any $x \in C$ and $\lambda > 0$. Then $f(x)^{\frac{1}{s}}$ is concave if and only if* $\log f(x)$ *is concave.*

*Proof.* One direction is trivial. If $f(x)^{\frac{1}{s}}$ is concave, then $\log f(x) = s \log(f(x)^{\frac{1}{s}})$ is concave since $\log(\cdot)$ is monotone and concave on $(0, +\infty)$. Conversely, if $\log f(x)$ is concave, then $f(\tau x + (1-\tau)y) \geq f(x)^{\tau} f(y)^{1-\tau}$, for any $x, y \in C, \tau \in [0,1]$. Now for any fixed $x, y \in C, \tau \in [0,1]$, we define $M = \tau f(x)^{\frac{1}{s}} + (1-\tau) f(y)^{\frac{1}{s}}$. We then have

$$\begin{aligned} f(\tau x + (1-\tau)y)^{\frac{1}{s}} &= f\left(\frac{\tau f(x)^{\frac{1}{s}}}{M}\frac{Mx}{f(x)^{\frac{1}{s}}} + \frac{(1-\tau) f(y)^{\frac{1}{s}}}{M}\frac{My}{f(y)^{\frac{1}{s}}}\right)^{\frac{1}{s}} \\ &\geq f\left(\frac{Mx}{f(x)^{\frac{1}{s}}}\right)^{\frac{\tau f(x)^{\frac{1}{s}}}{M}\cdot\frac{1}{s}} f\left(\frac{My}{f(y)^{\frac{1}{s}}}\right)^{\frac{(1-\tau) f(y)^{\frac{1}{s}}}{M}\cdot\frac{1}{s}} \\ &= (M^s)^{\frac{\tau f(x)^{\frac{1}{s}}}{M}\cdot\frac{1}{s} + \frac{(1-\tau) f(y)^{\frac{1}{s}}}{M}\cdot\frac{1}{s}} \\ &= M. \end{aligned}$$

Therefore $f(x)^{\frac{1}{s}}$ is concave. □

## 5.7 Other Results on $K$-trace

### 5.7.1 Some Corollaries

The following corollary follows from standard arguments on homogeneous, concave functions.

**Corollary 5.7.1.** *For any $p, q \in [0,1], p+q>0, s \in (0, \frac{1}{p+q}]$, and any $K \in \mathbb{C}^{n\times m}$, the function*

$$(A,B) \longmapsto \phi\big((B^{\frac{q}{2}}K^*A^pKB^{\frac{q}{2}})^s\big)^{\frac{1}{s(p+q)}} \tag{5.67}$$

*is jointly concave on $\mathbf{H}_n^+ \times \mathbf{H}_m^+$. For any $H \in \mathbf{H}_n$ and any $\{p_j\}_{j=1}^m \subset [0,1]$ such that $0 < \sum_{j=1}^m p_j \le 1$, the function*

$$(A^{(1)}, A^{(2)}, \ldots, A^{(m)}) \longmapsto \phi\big(\exp\big(H + \sum_{j=1}^m p_j \log A^{(j)}\big)\big)^{\frac{1}{\sum_{j=1}^m p_j}}, \tag{5.68}$$

*is jointly concave on $(\mathbf{H}_n^{++})^{\times m}$.*

*Proof.* Consider any matrix function $F : \mathbf{H}_n^+ \to [0,+\infty)$(or $\mathbf{H}_n^{++} \to [0,+\infty)$) that is positively homogeneous of order 1, i.e. $F(\lambda A) = \lambda F(A), \forall \lambda \ge 0$. By Lemma 5.6.9, we have that $F$ is concave $\Longleftrightarrow F^r$ is concave for some $r \in (0,1]$.

One can easily check that the functions (5.67) and (5.68) are positively homogeneous of order 1. Then this corollary follows from Theorem 5.3.3, Theorem 5.3.4 and Lemma 5.6.9. □

The following corollary is an analog of the concave part of Lemma 3.1 in [27].

**Corollary 5.7.2.** *For any $r \in [0,1], s \in [0, \frac{1}{r}]$ and any $\{K^{(j)}\}_{j=1}^m \subset \mathbb{C}^{n\times n}$, the function*

$$(A^{(1)}, A^{(2)}, \ldots, A^{(m)}) \longmapsto \phi\Big(\Big(\sum_{j=1}^m (K^{(j)})^*(A^{(j)})^r K^{(j)}\Big)^s\Big) \tag{5.69}$$

*is jointly concave on $(\mathbf{H}_n^+)^{\times m}$.*

*Proof.* Define

$$\widehat{A} = \begin{pmatrix} A^{(1)} & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & A^{(1)} & \ldots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \ldots & A^{(m)} \end{pmatrix} \in \mathbf{H}_{mn}^+, \quad \widehat{K} = \begin{pmatrix} K^{(1)} & \mathbf{0} & \ldots & \mathbf{0} \\ K^{(2)} & \mathbf{0} & \ldots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ K^{(m)} & \mathbf{0} & \ldots & \mathbf{0} \end{pmatrix} \in \mathbb{C}^{mn\times mn}.$$

Then we have

$$(\widehat{K}^*\widehat{A}^r\widehat{K})^s = \begin{pmatrix} \Big(\sum_{j=1}^m (K^{(j)})^*(A^{(j)})^r K^{(j)}\Big)^s & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \mathbf{0} \end{pmatrix},$$

and thus

$$\phi((\widehat{K}^*\widehat{A}^r\widehat{K})^s) = \phi\Big(\Big(\sum_{j=1}^m (K^{(j)})^*(A^{(j)})^r K^{(j)}\Big)^s\Big).$$

By Lemma 5.3.2, the left hand side above is concave in $\widehat{A}$, therefore the right hand side is jointly concave in $(A^{(1)}, A^{(2)}, \dots, A^{(m)})$. □

### 5.7.2 Multivariate Golden–Thompson Inequality

Sutter et al. [119] recently applied the operator interpolation in Theorem 5.6.7 to derive a multivariate extension of the Golden–Thompson (GT) inequality, which covers the original GT inequality and its three-matrix extension by Lieb [67].

Following the ideas in [119], we may also use Lemma 5.6.8 to further extend the multivariate GT inequality to a $k$-trace form. In what follows, we write $\prod_{j=1}^m X^{(j)}$ for the matrix multiplication in the index order, i.e. $\prod_{j=1}^m X^{(j)} = X^{(1)}X^{(2)}\cdots X^{(m)}$. We first present an analog of Theorem 3.2 in [119].

**Lemma 5.7.3.** *For any $A^{(1)}, A^{(2)}, \dots, A^{(m)} \in \mathbf{H}_n^+$, $p \in [1, +\infty)$, $r \in (0, 1]$, the following inequality holds:*

$$\log\phi\Big(\Big|\prod_{j=1}^m (A^{(j)})^r\Big|^{\frac{p}{r}}\Big) \le \int_{-\infty}^{+\infty} dt\beta_r(t)\log\phi\Big(\Big|\prod_{j=1}^m (A^{(j)})^{1+it}\Big|^p\Big). \tag{5.70}$$

*Proof.* Define

$$G(z) = \prod_{j=1}^m (A^{(j)})^z, \quad z \in \mathcal{S},$$

where $\mathcal{S}$ is defined as in Lemma 5.6.8. One can check that $G(z)$ is holomorphic in the interior of $\mathcal{S}$ and continuous on the boundary, and $\|G(z)\|$ is uniformly bounded on $\mathcal{S}$. We may first assume that each $A^{(j)} \in \mathbf{H}_n^{++}$ so that $(A^{(j)})^{it}, t \in \mathbb{R}$ is unitary. The result for $A^{(j)} \in \mathbf{H}_n^+$ can be obtained by continuity. Thus $G(it)$ is unitary for all $t \in \mathbb{R}$, and $|G(it)|^{p_0} = I_n$ for all $p_0$. Thus we can apply inequality (5.64) with $\theta = r, p_0 \to +\infty, p_1 = p, p_\theta = \frac{p}{r}$ to obtain

$$\log\phi(|G(r)|^{\frac{p}{r}}) \le \int_{-\infty}^{+\infty} dt\beta_r(t)\log\phi(|G(1+it)|^p),$$

which is exactly (5.70). □

Using a multivariate version of the Lie product formula, we immediately obtain the following from Lemma 5.7.3.

**Theorem 5.7.4** (Multivariate Golden–Thompson Inequality for k-trace)**.** *For any $A^{(1)}, A^{(2)}, \ldots, A^{(m)} \in \mathbf{H}_n$, the following inequality holds:*

$$\log \phi\Big(\big(\exp\big(\sum_{j=1}^{m} A^{(j)}\big)\big)^p\Big) \leq \int_{-\infty}^{+\infty} dt\, \beta_0(t) \log \phi\Big(\Big|\prod_{j=1}^{m} \exp\left((1+it)A^{(j)}\right)\Big|^p\Big). \quad (5.71)$$

*Proof.* We only need to replace $A^{(j)}$ in inequality (5.70) by $\exp(A^{(j)})$, and take $r \to 0$. Since each $\|\exp((1+it)A^{(j)})\| = \|\exp(A^{(j)})\exp(itA^{(j)})\| = \|\exp(A^{(j)})\|$ is uniformly bounded for all $t \in \mathbb{R}$, the right hand side of (5.70) then becomes the right hand side of (5.71). By a multivariate Lie product formula (see, e.g., [119])

$$\lim_{r \searrow 0} \Big(\exp(rX^{(1)})\exp(rX^{(2)})\cdots\exp(rX^{(m)})\Big)^{\frac{1}{r}} = \exp\big(\sum_{i=1}^{m} X^{(j)}\big),$$

the left hand side of (5.70) then becomes

$$\begin{aligned}
&\lim_{r \searrow 0} \log \phi\Big(\Big|\prod_{j=1}^{m} \exp\left(rA^{(j)}\right)\Big|^{\frac{p}{r}}\Big) \\
=\ &\lim_{r \searrow 0} \log \phi\Big(\Big(\prod_{j=1}^{m} \exp\left(rA^{(m-j+1)}\right)\prod_{j=1}^{m} \exp\left(rA^{(j)}\right)\Big)^{\frac{p}{2r}}\Big) \\
=\ &\log \phi\Big(\big(\exp\big(\sum_{j=1}^{m} 2A^{(j)}\big)\big)^{\frac{p}{2}}\Big) \\
=\ &\log \phi\Big(\big(\exp\big(\sum_{j=1}^{m} A^{(j)}\big)\big)^{p}\Big).
\end{aligned}$$

□

If we choose $m = 2, p = 2$ in Theorem 5.7.4 and replace $A^{(j)}$ by $\frac{1}{2}A^{(j)}$, the right hand side of inequality (5.71) is independent of $t$ due to the cyclicity of $\phi$. We then recover the $k$-trace GT inequality

$$\phi\big(\exp(A^{(1)} + A^{(2)})\big) \leq \phi\big(\exp(A^{(1)})\exp(A^{(2)})\big),$$

that we have obtained in Lemma 5.4.4. If we choose $m = 3, p = 2$ in Theorem 5.7.4 and again replace $A^{(j)}$ by $\frac{1}{2}A^{(j)}$, we have

$$\begin{aligned}
&\log \phi(\exp(A^{(1)} + A^{(2)} + A^{(3)})) \\
\leq &\int_{-\infty}^{+\infty} dt \beta_0(t) \log \phi\Big(\exp(A^{(1)}) \exp\big(\frac{1+it}{2}A^{(2)}\big) \exp(A^{(3)}) \exp\big(\frac{1-it}{2}A^{(2)}\big)\Big) \\
\leq &\log \phi\left(\int_{-\infty}^{+\infty} dt \beta_0(t) \exp(A^{(1)}) \exp\big(\frac{1+it}{2}A^{(2)}\big) \exp(A^{(3)}) \exp\big(\frac{1-it}{2}A^{(2)}\big)\right)
\end{aligned}$$

The second inequality above is due to concavity of logarithm and $\phi$. If we define

$$\mathcal{T}_A[B] = \int_0^{+\infty} dt (A + tI_n)^{-1} B (A + tI_n)^{-1}, \quad A, B \in \mathbf{H}_n^{++},$$

and use Lemma 3.4 in [119] that

$$\int_0^{+\infty} dt (A^{-1} + tI_n)^{-1} B (A^{-1} + tI_n)^{-1} = \int_{-\infty}^{+\infty} dt \beta_0(t) A^{\frac{1+it}{2}} B A^{\frac{1-it}{2}}, \quad A, B \in \mathbf{H}_n^{++},$$

we then further obtain

$$\phi(\exp(A^{(1)} + A^{(2)} + A^{(3)})) \leq \phi(\exp(A^{(1)}) \mathcal{T}_{\exp(-A^{(2)})}[\exp(A^{(3)})]).$$

This can be seen as a $k$-trace generalization of Lieb's [67] three-matrix extension of the GT inequality that $\mathrm{Tr}[\exp(A + B + C)] \leq \mathrm{Tr}[\exp(A) \mathcal{T}_{\exp(-B)}[\exp(C)]]$.

### 5.7.3 Monotonicity Preserving and Concavity Preserving

As mentioned in Section 5.4.3, Loewner's theorem says that a real-valued function $f$ on $(0, +\infty)$ is operator monotone if and only if it admits an analytic continuation to a Herglotz function. Therefore, the extension of a monotone scalar function to Hermitian matrices is not necessarily operator monotone. For instance, let $f(x) = x^3$, and

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix},$$

then $f$ is monotone increasing, and $A \preceq B$. But neither $A^3 \preceq B^3$ nor $A^3 \succeq B^3$ is true. However, a composition with trace will preserve the monotonicity. That is, $\mathrm{Tr}[f(A)]$ is monotone increasing (or decreasing) in $A$ with respect to Loewner partial order, if $f$ is monotone increasing (or decreasing). Likewise, if $f$ is concave (or convex), its extension to Hermitian matrices is not necessarily operator concave (or convex), but $A \mapsto \mathrm{Tr}[f(A)]$ is still concave (or convex). One can see Theorem 2.10 in [24]. This means that, some partial information like trace may preserve

monotonicity and concavity. In fact, we will show that for any integer $k$ the partial information $\phi(\cdot) = \mathrm{Tr}_k[\cdot]^{\frac{1}{k}}$ also preserves monotonicity and concavity of scalar functions. But we need to restrict to $f$ that only takes values in $[0, +\infty)$. We need the following lemma for proving concavity preserving.

**Lemma 5.7.5.** *For any $A \in \mathbf{H}_n^+$, let* $\mathrm{diag}(A)$ *denote the diagonal part of $A$, then*

$$\phi(A) \leq \phi(\mathrm{diag}(A)).$$

*Proof.* Let $D$ be a $n \times n$ diagonal matrix, whose diagonal entries follow independent Rademacher distributions, i.e.

$$D_{ii} = \begin{cases} 1, & \text{with prob. } 0.5, \\ -1, & \text{with prob. } 0.5. \end{cases}$$

Since $D^2 \equiv I_n$, we have $\phi(A) = \phi(DAD)$. Also notice that $\mathbb{E}[DAD] = \mathrm{diag}(A)$, since $\mathbb{E}[D_{ii}D_{jj}] = \delta_{ij}$. Then by concavity of $\phi$, we have

$$\phi(A) = \mathbb{E}\phi(DAD) \leq \phi(\mathbb{E}[DAD]) = \phi(\mathrm{diag}(A)).$$

□

Lemma 5.7.5 can be proved, instead, using the concept of Majorization. Let $\boldsymbol{a} = \{a_i\}_i^n$ and $\boldsymbol{b} = \{b_i\}_i^n$ be two sequences, both in descending order, i.e. $a_1 \geq a_2 \geq \cdots \geq a_n$, $b_1 \geq b_2 \geq \cdots \geq b_n$. We say $\boldsymbol{b}$ majorizes $\boldsymbol{a}$, denoted by $\boldsymbol{b} \succeq \boldsymbol{a}$, if

$$\sum_{i=1}^{k} b_i \geq \sum_{i=1}^{k} a_i, \quad 1 \leq k \leq n,$$

and the equality holds for $k = n$. It is not hard to show that, if $\boldsymbol{b} \succeq \boldsymbol{a}$, and $\boldsymbol{a}, \boldsymbol{b} \subset [0, +\infty)$, then $\prod_{i=1}^{d} b_i \leq \prod_{i=1}^{d} a_i$. Now for any $A \in \mathbf{H}_n^+$, let $\boldsymbol{\lambda} = \{\lambda_i\}_{i=1}^n$ and $\boldsymbol{a} = \{a_i\}_i^n$ be the eigenvalues and the diagonal entries of $A$ respectively, both in descending order. Then since

$$\sum_{i=1}^{k} \lambda_i = \max_{\substack{\{v_i\}_{i=1}^k \subset \mathbb{C}^n \\ v_i^* v_j = \delta_{ij}}} \sum_{i=1}^{k} v_i^* A v_i \geq \sum_{i=1}^{k} e_i^* A e_i = \sum_{i=1}^{k} a_i, \quad 1 \leq k < n$$

and $\sum_{i=1}^n \lambda_i = \mathrm{Tr}[A] = \sum_{i=1}^n a_i$, we have $\boldsymbol{\lambda} \succeq \boldsymbol{a}$. Therefore we know that

$$\det[A] = \prod_{i=1}^{n} \lambda_i \leq \prod_{i=1}^{n} a_i = \det[\mathrm{diag}(A)].$$

Then using the equivalent definition (5.13) of $k$-trace, we have that, for any $1 \leq k \leq n$,

$$\begin{aligned}\mathrm{Tr}_k[A] &= \sum_{1\leq i_1<i_2<\cdots<i_k\leq n} \det[A_{(i_1\cdots i_k, i_1\cdots i_k)}] \\ &\leq \sum_{1\leq i_1<i_2<\cdots<i_k\leq n} A_{i_1i_1}A_{i_2i_2}\cdots A_{i_ki_k}, \\ &= \mathrm{Tr}_k[\mathrm{diag}(A)].\end{aligned}$$

**Theorem 5.7.6.** *Given a function $f : [0, +\infty) \to [0, +\infty)$, if $f$ is monotone increasing (or decreasing) as a scalar function, then $\phi(f(\cdot))$ is monotone increasing (or decreasing) on $\mathbf{H}_n^+$, in the sense that $\phi(f(A)) \geq \phi(f(B))$(or $\phi(f(A)) \leq \phi(f(B))$) if $A, B \in \mathbf{H}_n^+, A \geq B$; if $f$ is concave as a scalar function, then $\phi(f(\cdot))$ is concave on $\mathbf{H}_n^+$.*

*Proof.* We first prove the monotonicity preserving property of $\phi$. For any matrix $A \in \mathbf{H}_n$, we denote by $\lambda_i(A)$ the $i_{\text{th}}$ largest eigenvalue of $A$. For any $A, B \in \mathbf{H}_n^+$, if $A \geq B$, then $\lambda_i(A) \geq \lambda_i(B), 1 \leq i \leq n$. Therefore if $f$ is monotone increasing, we immediately have

$$\lambda_i(f(A)) = f(\lambda_i(A)) \geq f(\lambda_i(B)) = \lambda_i(f(B)), \quad 1 \leq i \leq n,$$

and thus $\phi(f(A)) \geq \phi(f(B))$ by definition $\phi$. Similarly, if $f$ is monotone decreasing, we have

$$\lambda_i(f(A)) = f(\lambda_{n-i+1}(A)) \leq f(\lambda_{n-i+1}(B)) = \lambda_i(f(B)), \quad 1 \leq i \leq n,$$

and thus $\phi(f(A)) \leq \phi(f(B))$.

Next we prove the concavity preserving property of $\phi$. Given any $A, B \in \mathbf{H}_n^+$, and any $\tau \in [0, 1]$, we define $C = \tau A + (1 - \tau)B$. Let $U = [u_1, u_2, \cdots, u_n]$ be a unitary matrix such that the columns are all eigenvectors of $C$, then

$$U^* f(C)U = f(U^*CU) = f(\mathrm{diag}(U^*CU)).$$

If $f$ is concave, then

$$f(u_i^*Cu_i) = f(\tau u_i^*Au_i + (1-\tau)u_i^*Bu_i) \geq \tau f(u_i^*Au_i) + (1-\tau)f(u_i^*Bu_i), \quad 1 \leq i \leq n,$$

and thus $f(\mathrm{diag}(U^*CU)) \geq \tau f(\mathrm{diag}(U^*AU)) + (1-\tau) f(\mathrm{diag}(U^*BU))$. Further, for any unit vector $u \in \mathbb{C}^n$, we have

$$\begin{aligned} f(u^*Au) &= f\Big(\sum_{i=1}^n \lambda_i(A) u^* v_i v_i^* u\Big) = f\Big(\sum_{i=1}^n \lambda_j(A) |v_i^* u|^2\Big) \\ &\geq \sum_{i=1}^n |v_i^* u|^2 f(\lambda_i(A)) = u^*\Big(\sum_{i=1}^n f(\lambda_i(A)) v_i v_i^*\Big) u = u^* f(A) u, \end{aligned}$$

where $v_1, v_2, \cdots, v_n$ are all eigenvectors of $A$, and we have used that $\sum_{i=1}^n |v_i^* u|^2 = \|u\|_2^2 = 1$. Then we have $f(\mathrm{diag}(U^*AU)) \geq \mathrm{diag}(U^* f(A) U)$. Similar, we have $f(\mathrm{diag}(U^*BU)) \geq \mathrm{diag}(U^* f(B) U)$. Finally, we can compute

$$\begin{aligned} \phi(f(C)) &= \phi(U^* f(C) U) \\ &= \phi\big(f(\mathrm{diag}(U^*CU))\big) \\ &\geq \phi\big(\tau f(\mathrm{diag}(U^*AU)) + (1-\tau) f(\mathrm{diag}(U^*BU))\big) \\ &\geq \tau\phi\big(f(\mathrm{diag}(U^*AU))\big) + (1-\tau)\phi\big(f(\mathrm{diag}(U^*BU))\big) \\ &\geq \tau\phi\big(\mathrm{diag}(f(U^*AU))\big) + (1-\tau)\phi\big(\mathrm{diag}(f(U^*BU))\big) \\ &\geq \tau\phi\big(f(U^*AU)\big) + (1-\tau)\phi\big(f(U^*BU)\big) \\ &= \tau\phi\big(U^* f(A) U\big) + (1-\tau)\phi\big(U^* f(B) U\big) \\ &= \tau\phi\big(f(A)\big) + (1-\tau)\phi\big(f(B)\big). \end{aligned}$$

We have used Lemma 5.7.5 for the last inequality above. Therefore $\phi(f(\cdot))$ is concave on $\mathbf{H}_n^+$. □

*Chapter 6*

# CONCLUDING DISCUSSIONS

Problems related to PD matrices have aroused great interest in many fields of science. In this thesis, we developed and discussed theories and algorithms for a general class of PD matrices on three main problems: solving a PD linear system, computing the eigenpairs of a PD matrices, and studying the concentration of random PD matrices. Our methods integrate the ideas of a variety of existing methodologies and show to be superior in many particular examples.

The problem of approximating the inverse of a PD operator with localized basis functions is important in both the physical and data sciences. To pursue this matter, we propose in the first part of the thesis an operator compression framework based on the notion of energy decomposition. The energy decomposition $A = \sum_{k=1}^{m} E_k$ extracts the hidden topological and geometric information of a PD matrix $A$, which serves the purpose of finding an adaptive partitioning $\mathcal{P}$ of the finest underlying structure. Specifically, we introduce two important local measurements, the error factor $\varepsilon(\mathcal{P})$ and the condition factor $\delta(\mathcal{P})$, to provide rigorous a priori estimate estimates for our partitioning technique. These two factors can be calculated efficiently by solving local and partial eigen problems and therefore are practically useful in designing our algorithms. Upon the establishment of an appropriate partitioning, we follow the ideas of the modified coarse space by Målqvist and Peterseim [75] and the gamblet transform by Owhadi [88] to construct an effective coarse level basis $\Psi$ consisting of basis functions with exponentially decaying profiles. The exponential decay property of $\Psi$ enables us to replace it with a localized basis $\widetilde{\Psi}$ without compromising the expected compression accuracy; and because of this the compressed operator preserves the intrinsic sparsity of the original operator $A$, ensuring the low complexity of our approach in practice. With all the preceding ingredients, we propose a nearly-linear time algorithm to obtain an appropriate partitioning and a localized basis for compressing the operator with prescribed accuracy and bounded condition number.

Having a generic operator compression framework, we follow the idea in [88] to extend the compression scheme hierarchically to form a MMD algorithm. The main

idea is to decompose the operator into multiple scales of resolution,

$$A^{-1} = \sum_{k=1}^{K} \mathcal{U}^{(k)}((\mathcal{U}^{(k)})^T A \mathcal{U}^{(k)})^{-1}(\mathcal{U}^{(k)})^T + \mathbf{\Psi}^{(K)}((\mathbf{\Psi}^{(K)})^T A \mathbf{\Psi}^{(K)})^{-1}(\mathbf{\Psi}^{(K)})^T,$$

such that the relative condition number in each scale can be bounded. By passing the energy decomposition from the finest level to the coarsest level, we perform our partitioning and basis construction techniques level-by-level in a recursive manner. This provides flexibility and convenience to deal with various, and even unknown multiresolution behavior appearing in the matrix. Our MMD method further leads us to develop a nearly-linear time solver for large and sparse PD systems up to an error $\epsilon$ with time complexity

$$O(m \log n \cdot \left(\log \epsilon^{-1} + \log n\right)^c \log \epsilon^{-1}),$$

where $m$ is the number of nonzero entries of $A$ and $c$ is some absolute constant depending only on the geometric property of $A$.

Our groundwork introduces the idea of energy decomposition and its applications in operator compression and solving PD 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. For instance, 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 $\mathcal{P}$ 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 application-wise planning, we believe that our energy decomposition framework can be specifically modified to suit the purpose of solving elliptic PDEs with high contrast coefficients, as demonstrated in the high-contrast problem in Section 2.4.2. Based on our energy decomposition, more in-depth analysis and improvement could be made to show that such framework is one of the possible candidates to solve the elliptic type problem with highly varying coefficients. Moreover, regarding the partitioning procedure and the locality of the basis in our algorithms, the localized MMD solver can be further improved to fit into the needs of frequent updating of the solver. For example, in 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 updates size is small.

Following the preceding MMD framework, we propose a hierarchically preconditioned eigensolver to compute a relatively large number of leftmost eigenpairs of a sparse PD matrix. This eigensolver exploits the well-conditioned property of the decomposition components obtained through the MMD, the nice converging property of the IRLM [64], and also the preconditioning characteristics of the CG method. In particular, we develop an extension-refinement iterative scheme, in which eigenpairs are hierarchically extended and refined from the ones obtained from the previous level up to a desired amount and a prescribed accuracy. To find new eigenvector candidate in the complement space, we need to solve linear systems with respect to a compressed operator $A_{st}$ in each level. For this purpose, we introduce a specially designed spectrum-preserving preconditioner $M$ that corrects the orthogonality of the eigenvectors of the compressed operator, yielding a small restricted condition number for using the PCG method. Based on the design of our method, we present theoretical analysis on its runtime complexity and asymptotic behavior. We have also conducted quantitative numerical experiments and performance comparison with the IRLM, which demonstrated the efficiency and effectiveness of our proposed algorithm. The results show that our preconditioning technique remarkably reduces the number of iteration in each call of the PCG method.

We remark that the proposed algorithm and its implementation are still in the early stage as the main purpose of this work is to explore the possibility of integrating the multiresolution operator compression framework with the Krylov-type iterative eigensolver. Therefore, one of the future topics is to conduct a comprehensive numerical studies of our algorithm to various large-scale, real data such as graph Laplacians of real network data, or stiffness matrices stemmed from the discretization of high-contrasted elliptic PDEs. These studies will help numerically confirm the asymptotic behavior of the relative condition numbers of $M$ and $A_{st}$, especially when we need to compute a large number of leftmost eigenpairs from large-scale operators.

Another possible research direction is to investigate the parallelization of our algorithm. This is important when we solve a large-scale eigenvalue problem. One way to implement parallelization is by modifying the underlying Lanczos method into a block version. Based on the conventional Lanczos iteration, many researchers

have proposed block Lanczos algorithms [33, 42] for computing the leading eigenpairs of symmetric matrices. Instead of computing only one candidate vector, a Lanczos method applies the target operator to a bunch of vectors in each iteration and hence expands the Krylov subspace much faster. We may also incorporate this block technique into our eigensolver and hence reduce its time complexity if parallel computation is available.

In the last part of the thesis, we turn to a more theoretical project on developing concentration inequalities on eigenvalue sums. To obtain bounds on partial sums of eigenvalues, we introduce the notion of $k$-trace and discuss its properties. With the help of the $k$-trace, we show that the sum of the $k$ largest (or smallest) eigenvalues of a general class of random PD matrices also obeys a Chernoff-type tail bound, generalizing the existing estimates on the largest (or smallest) eigenvalue [122]. These new estimates provides theoretical guarantees for randomized algorithms in eigenvalue-related problems, in particular for random spectral sparsification of PD matrices. Our concentration results are consequences of a generalized Lieb's concavity theorem that the map

$$(A, B) \longmapsto \mathrm{Tr}_k \left[ (B^{\frac{q}{2}} K^* A^p K B^{\frac{q}{2}})^s \right]^{\frac{1}{k}}$$

is jointly concave on $\mathbf{H}_n^+ \times \mathbf{H}_m^+$ for any $p, q \in [0, 1]$, $s \in [0, \frac{1}{p+q}]$ and any $K \in \mathbb{C}^{n \times m}$, which extends the classic Lieb's concavity theorem [67] form the normal trace to $k$-trace functions. Our proof of the generalized Lieb's concavity theorem relies on the properties of $k$-trace functions and the use of an operator interpolation technique due to Stein [116].

We would like to remark that our generalized Lieb's concavity theorem (Theorem 5.3.3) can be further extended to a more general class of functions. We say a function $\phi : \mathbf{H}_n^+ \to \mathbb{R}$ is a monotone concave symmetric form, if

(i) $\phi(U^* A U) = \phi(A)$ for any $A \in \mathbf{H}_n^+$ and unitary matrix $U \in \mathbb{C}^{n \times n}$,

(ii) $A \preceq B$ implies $\phi(A) \leq \phi(B)$ for any $A, B \in \mathbf{H}_n^+$, and

(iii) $\phi(\tau A + (1-\tau)B) \geq \tau\phi(A) + (1-\tau)\phi(B)$ for any $A, B \in \mathbf{H}_n^+$ and any $\tau \in [0, 1]$.

It has been proved by the author of this thesis [54] that for any monotone concave symmetric form $\phi : \mathbf{H}_n^+ \to \mathbb{R}$, the map

$$(A, B) \longmapsto \phi \left( (B^{\frac{q}{2}} K^* A^p K B^{\frac{q}{2}})^s \right)$$

is jointly concave on $\mathbf{H}_n^+ \times \mathbf{H}_m^+$ for all $p, q \in [0, 1]$, $s \in [0, \frac{1}{p+q}]$ and all $K \in \mathbb{C}^{n\times m}$. Notice that the $k$-trace function $\mathrm{Tr}_k[\cdot]^{\frac{1}{k}}$ is a monotone concave symmetric form for all $1 \leq k \leq n$. Therefore the more general result in [54] covers our Theorem 5.3.3.

As we have mentioned earlier, it is possible to extend our concentration inequalities on the sum of the largest (or smallest) eigenvalues to analogous results on the sum of arbitrary successive eigenvalues. In fact, Tropp and Gittens [40] have established concentration inequalities on arbitrary single eigenvalue by projecting the random matrix onto a subspace. We may therefore combine our $k$-trace methods and their subspace argument to obtain concentration results on the sum of arbitrary successive eigenvalues.

Apart from its particular use in proving our concentration inequalities, the $k$-trace is also of theoretical interest by itself, as it has many interpretations corresponding to different aspects of matrix theories. For instance, the $k$-trace has a direct connection to theories of mixed discriminants and anti-symmetric matrix tensors. However, so far we have not found another specific example in which an explicit use of the $k$-trace is crucial. As for future projects, we want to explore more theoretical applications of the $k$-trace, which may extend the corresponding theories on matrix trace functions to an even broader scope.

# BIBLIOGRAPHY


[1] W. Aiello, F. Chung, and L. Lu. A random graph model for massive graphs. In *Proceedings of the Thirty-second Annual ACM Symposium on Theory of Computing*, STOC '00, pages 171–180, New York, NY, USA, 2000. ACM.

[2] A. Aleksandrov. On the theory of mixed volumes of convex bodies. IV. mixed discriminants and mixed volumes. *Mat. Sb.(NS)*, 3(1):938, 1938.

[3] T. Ando. Concavity of certain maps on positive definite matrices and applications to Hadamard products. *Linear Algebra and its Applications*, 26:203–241, 1979.

[4] H. Araki. On an inequality of Lieb and Thirring. *Letters in Mathematical Physics*, 19(2):167–170, 1990.

[5] J. Ashburner. A fast diffeomorphic image registration algorithm. *Neuroimage*, 38(1):95–113, 2007.

[6] R. Bapat. Mixed discriminants of positive semidefinite matrices. *Linear Algebra and its Applications*, 126:107–124, 1989.

[7] 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, 1994.

[8] K.-J. Bathe and E. L. Wilson. *Numerical methods in finite element analysis*, volume 197. Prentice-Hall Englewood Cliffs, NJ, 1976.

[9] M. Belkin, I. Matveeva, and P. Niyogi. Regularization and semi-supervised learning on large graphs. In *International Conference on Computational Learning Theory*, pages 624–638. Springer, 2004.

[10] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *NIPS*, volume 14, pages 585–591, 2001.

[11] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of machine learning research*, 7(Nov):2399–2434, 2006.

[12] L. Bergamaschi and E. Bozzo. Computing the smallest eigenpairs of the graph Laplacian. *SeMA Journal*, pages 1–16, 2015.

[13] L. Bergamaschi, G. Gambolati, and G. Pini. Asymptotic convergence of conjugate gradient methods for the partial symmetric eigenproblem. *Numerical linear algebra with applications*, 4(2):69–84, 1997.

[14] R. Bishop and S. Goldberg. *Tensor Analysis on Manifolds*. Dover Books on Mathematics. Dover Publications, 1980.

[15] F. L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on pattern analysis and machine intelligence*, 11(6):567–585, 1989.

[16] Y. Boykov and V. Kolmogorov. Computing geodesics and minimal surfaces via graph cuts. In *ICCV*, volume 3, pages 26–33, 2003.

[17] E. Bozzo and M. Franceschet. Effective and efficient approximations of the generalized inverse of the graph Laplacian matrix with an application to current-flow betweenness centrality. *arXiv preprint arXiv:1205.4894*, 2012.

[18] E. Bozzo and M. Franceschet. Resistance distance, closeness, and betweenness. *Social Networks*, 35(3):460–469, 2013.

[19] S. C. Brenner and C. Carstensen. Finite element methods. *Encyclopedia of computational mechanics*, 2004.

[20] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

[21] D. Cai, X. He, J. Han, and T. S. Huang. Graph regularized nonnegative matrix factorization for data representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1548–1560, 2011.

[22] D. Calvetti, L. Reichel, and D. C. Sorensen. An implicitly restarted Lanczos method for large symmetric eigenvalue problems. *Electronic Transactions on Numerical Analysis*, 2(1):21, 1994.

[23] Y. Cao, M. I. Miller, R. L. Winslow, and L. Younes. Large deformation diffeomorphic metric mapping of vector fields. *IEEE transactions on medical imaging*, 24(9):1216–1230, 2005.

[24] E. Carlen. Trace inequalities and quantum entropy: an introductory course. *Entropy and the quantum*, 529:73–140, 2010.

[25] E. A. Carlen, R. L. Frank, and E. H. Lieb. Some operator and trace function convexity theorems. *Linear Algebra and its Applications*, 490:174–185, 2016.

[26] E. A. Carlen, R. L. Frank, and E. H. Lieb. Inequalities for quantum divergences and the Audenaert–Datta conjecture. *Journal of Physics A: Mathematical and Theoretical*, 51(48):483001, 2018.

[27] E. A. Carlen and E. H. Lieb. A Minkowski type trace inequality and strong subadditivity of quantum entropy II: Convexity and concavity. *Letters in Mathematical Physics*, 83(2):107–126, Feb 2008.

[28] T. F. Chan and J. Zou. Additive Schwarz domain decomposition methods for elliptic problems on unstructured meshes. *Numerical Algorithms*, 8(2):329–346, 1994.

[29] K. Chaudhuri, F. Chung, and A. Tsiatas. Spectral clustering of graphs with general degrees in the extended planted partition model. In *Conference on Learning Theory*, pages 35–1, 2012.

[30] F. R. Chung. *Spectral graph theory*, volume 92. American Mathematical Soc., 1997.

[31] S. Cocco, R. Monasson, and M. Weigt. From principal component to direct coupling analysis of coevolution in proteins: Low-eigenvalue modes are needed for structure prediction. *PLoS Computational Biology*, 9(8):e1003176, 2013.

[32] C. Colovos and T. O. Yeates. Verification of protein structures: Patterns of nonbonded atomic interactions. *Protein science*, 2(9):1511–1519, 1993.

[33] J. Cullum and W. E. Donath. A block Lanczos algorithm for computing the q algebraically largest eigenvalues and a corresponding eigenspace of large, sparse, real symmetric matrices. In *1974 IEEE Conference on Decision and Control including the 13th Symposium on Adaptive Processes*, pages 505–509. IEEE, 1974.

[34] A. d'Aspremont, L. El Ghaoui, M. I. Jordan, and G. R. Lanckriet. A direct formulation for sparse PCA using semidefinite programming. *SIAM review*, 49(3):434–448, 2007.

[35] W. F. Donoghue. Monotone matrix functions and analytic continuation. 1974.

[36] A. Ebadian, I. Nikoufar, and M. E. Gordji. Perspectives of matrix convex functions. *Proceedings of the National Academy of Sciences*, 108(18):7313–7314, 2011.

[37] H. Epstein. Remarks on two theorems of E. Lieb. *Communications in Mathematical Physics*, 31(4):317–325, 1973.

[38] P. Erdős and A. Rényi. On random graphs I. *Publicationes Mathematicae (Debrecen)*, 6:290–297, 1959 1959.

[39] J. Francis. The transformation: a unitary analogue to the transformation. i. *Comput. J.*, 4:265–271, 1961.

[40] A. Gittens and J. A. Tropp. Tail bounds for all eigenvalues of a sum of random matrices. *arXiv preprint arXiv:1104.4513*, 2011.

[41] S. Goedecker. Low complexity algorithms for electronic structure calculations. *Journal of Computational Physics*, 118(2):261–268, 1995.

[42] R. G. Grimes, J. G. Lewis, and H. D. Simon. A shifted block Lanczos algorithm for solving sparse symmetric generalized eigenproblems. *SIAM Journal on Matrix Analysis and Applications*, 15(1):228–272, 1994.

[43] T. Guhr, A. Müller-Groeling, and H. A. Weidenmüller. Random-matrix theories in quantum physics: common concepts. *Physics Reports*, 299(4-6):189–425, 1998.

[44] W. Hackbusch. *Multi-grid methods and applications*, volume 4. Springer Science & Business Media, 2013.

[45] J. Hadamard et al. Théorème sur les séries entières. *Acta Mathematica*, 22:55–63, 1899.

[46] E. Heinz. Beiträge zur störungstheorie der spektralzerleung. *Mathematische Annalen*, 123(1):415–438, 1951.

[47] F. Hiai. Concavity of certain matrix trace and norm functions. *Linear Algebra and its Applications*, 439(5):1568–1589, 2013.

[48] F. Hiai. Concavity of certain matrix trace and norm functions. II. *Linear Algebra and its Applications*, 496:193–220, 2016.

[49] F. Hiai et al. Concavity of certain matrix trace functions. *Taiwanese Journal of Mathematics*, 5(3):535–554, 2001.

[50] F. Hiai, R. König, and M. Tomamichel. Generalized log-majorization and multivariate trace inequalities. In *Annales Henri Poincaré*, volume 18, pages 2499–2521. Springer, 2017.

[51] I. I. Hirschman. A convexity theorem for certain groups of transformations. *Journal d'Analyse Mathématique*, 2(2):209–218, 1952.

[52] T. Y. Hou, D. Huang, K. C. Lam, and P. Zhang. An adaptive fast solver for a general class of positive definite matrices via energy decomposition. *Multiscale Modeling & Simulation*, 16(2):615–678, 2018.

[53] Y. T. Hou and P. Zhang. Sparse operator compression of higher-order elliptic operators with rough coefficients. *Research in Mathematical Sciences, in press*, 2017.

[54] D. Huang. Generality of Lieb's concavity theorem. *arXiv preprint arXiv:1906.00002*, 2019.

[55] I. Jolliffe. *Principal component analysis*. Wiley Online Library, 2002.

[56] I. T. Jolliffe and J. Cadima. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202, 2016.

[57] 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*, pages 11–21. ACM, 2004.

[58] H. Kosaki. Interpolation theory and the Wigner–Yanase–Dyson–Lieb concavity. *Communications in Mathematical Physics*, 87(3):315–329, 1982.

[59] 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 on*, pages 590–598. IEEE, 2011.

[60] F. Kraus. Über konvexe matrixfunktionen. *Mathematische Zeitschrift*, 41(1):18–42, 1936.

[61] R. d. L. Kronig and W. G. Penney. Quantum mechanics of electrons in crystal lattices. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 130(814):499–513, 1931.

[62] L. Laloux, P. Cizeau, M. Potters, and J.-P. Bouchaud. Random matrix theory and financial correlations. *International Journal of Theoretical and Applied Finance*, 3(03):391–397, 2000.

[63] M. G. Larson and A. Målqvist. Adaptive variational multiscale methods based on a posteriori error estimation: energy norm estimates for elliptic problems. *Computer methods in applied mechanics and engineering*, 196(21):2313–2324, 2007.

[64] R. B. Lehoucq and D. C. Sorensen. Deflation techniques for an implicitly restarted Arnoldi iteration. *SIAM Journal on Matrix Analysis and Applications*, 17(4):789–821, 1996.

[65] R. B. Lehoucq, D. C. Sorensen, and C. Yang. *ARPACK users' guide: solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods*. SIAM, 1998.

[66] P. Li. The Alexandrov–Fenchel type inequalities, revisited. *arXiv preprint arXiv:1710.00520*, 2017.

[67] E. H. Lieb. Convex trace functions and the Wigner–Yanase–Dyson conjecture. *Advances in Mathematics*, 11(3):267–288, 1973.

[68] E. H. Lieb and M. B. Ruskai. Proof of the strong subadditivity of quantum-mechanical entropy. *Journal of Mathematical Physics*, 14(12):1938–1941, 1973.

[69] Q. Lin and H. Xie. A multi-level correction scheme for eigenvalue problems. *Mathematics of Computation*, 84(291):71–88, 2015.

[70] G. Lindblad. Entropy, information and quantum measurements. *Communications in Mathematical Physics*, 33(4):305–322, 1973.

[71] O. E. Livne and A. Brandt. Lean algebraic multigrid (LAMG): Fast graph Laplacian linear solver. *SIAM Journal on Scientific Computing*, 34(4):B499–B522, 2012.

[72] K. Löwner. Über monotone matrixfunktionen. *Mathematische Zeitschrift*, 38(1):177–216, 1934.

[73] F. Lust-Piquard. Inégalités de Khintchine dans $C_p(1 < p < \infty)$. *CR Acad. Sci. Paris*, 303:289–292, 1986.

[74] L. Mackey, M. I. Jordan, R. Y. Chen, B. Farrell, J. A. Tropp, et al. Matrix concentration inequalities via the method of exchangeable pairs. *The Annals of Probability*, 42(3):906–945, 2014.

[75] A. Målqvist and D. Peterseim. Localization of elliptic multiscale problems. *Mathematics of Computation*, 83(290):2583–2603, 2014.

[76] Á. Martínez. Tuned preconditioners for the eigensolution of large spd matrices arising in engineering problems. *Numerical Linear Algebra with Applications*, 23(3):427–443, 2016.

[77] M. L. Mehta. *Random matrices*, volume 142. Elsevier, 2004.

[78] L. Meirovitch. *Elements of vibration analysis*. McGraw-Hill, 1975.

[79] N. Mishra, R. Schreiber, I. Stanton, and R. E. Tarjan. Clustering social networks. In A. Bonato and F. R. K. Chung, editors, *Algorithms and Models for the Web-Graph*, pages 56–67, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[80] B. Moghaddam, Y. Weiss, and S. Avidan. Spectral bounds for sparse PCA: Exact and greedy algorithms. *Advances in neural information processing systems*, 18:915, 2006.

[81] A. Myronenko and X. Song. Point set registration: Coherent point drift. *IEEE transactions on pattern analysis and machine intelligence*, 32(12):2262–2275, 2010.

[82] M. Newman. *Networks: an introduction*. Oxford university press, 2010.

[83] M. E. Newman. Modularity and community structure in networks. *Proceedings of the national academy of sciences*, 103(23):8577–8582, 2006.

[84] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*, pages 849–856, 2002.

[85] I. Nikoufar, A. Ebadian, and M. E. Gordji. The simplest proof of Lieb concavity theorem. *Advances in Mathematics*, 248:531 – 533, 2013.

[86] R. I. Oliveira. Concentration of the adjacency matrix and of the Laplacian in random graphs with independent edges, 2009.

[87] R. I. Oliveira. Sums of random Hermitian matrices and an inequality by Rudelson. *Electronic Communications in Probability*, 15:203–212, 2010.

[88] H. Owhadi. Multigrid with rough coefficients and multiresolution operator decomposition from hierarchical information games. *SIAM Review*, 59(1):99–149, 2017.

[89] H. Owhadi and C. Scovel. Universal scalable robust solvers from computational information games and fast eigenspace adapted multiresolution analysis. *arXiv preprint arXiv:1703.10761*, 2017.

[90] V. Ozolinš, R. Lai, R. Caflisch, and S. Osher. Compressed modes for variational problems in mathematics and physics. *Proceedings of the National Academy of Sciences*, 110(46):18368–18373, 2013.

[91] A. A. Panov. On some properties of mixed discriminants. *Sbornik: Mathematics*, 56(2):279–293, 1987.

[92] J. Peetre. A theory of interpolation spaces. *Notes, Universidade de Brasilia*, 1963.

[93] D. Petz. A survey of certain trace inequalities. *Banach Center Publications*, 30(1):287–298, 1994.

[94] G. Pisier and Q. Xu. Non-commutative martingale inequalities. *Communications in mathematical physics*, 189(3):667–698, 1997.

[95] V. Plerou, P. Gopikrishnan, B. Rosenow, L. A. N. Amaral, T. Guhr, and H. E. Stanley. Random matrix approach to cross correlations in financial data. *Physical Review E*, 65(6):066126, 2002.

[96] T. Qin and K. Rohe. Regularized spectral clustering under the degree-corrected stochastic blockmodel. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3120–3128. Curran Associates, Inc., 2013.

[97] M. Riesz et al. Sur les maxima des formes bilinéaires et sur les fonctionnelles linéaires. *Acta mathematica*, 49(3–4):465–497, 1926.

[98] K. Rohe, S. Chatterjee, B. Yu, et al. Spectral clustering and the high-dimensional stochastic blockmodel. *The Annals of Statistics*, 39(4):1878–1915, 2011.

[99] E. Romero, M. B. Cruz, J. E. Roman, and P. B. Vasconcelos. A parallel implementation of the Jacobi–Davidson eigensolver for unsymmetric matrices. In *VECPAR*, pages 380–393. Springer, 2010.

[100] J. J. Rotman. *Advanced modern algebra; 2nd ed.* Graduate studies in mathematics. American Mathematical Society, Providence, RI, 2010.

[101] M. Rudelson. Random vectors in the isotropic position. *Journal of Functional Analysis*, 164(1):60–72, 1999.

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

[103] R. S. Sampath and G. Biros. A parallel geometric multigrid method for finite elements on octree meshes. *SIAM Journal on Scientific Computing*, 32(3):1361–1392, 2010.

[104] F. Schäfer, T. Sullivan, and H. Owhadi. Compression, inversion, and approximate pca of dense kernel matrices at near-linear computational complexity. *arXiv preprint arXiv:1706.02205*, 2017.

[105] R. Schneider. Convex bodies: The Brunn–Minkowski theory, second expanded edition. *Encyclopedia of Mathematics and its Applications*, 1(151):ALL–ALL, 2014.

[106] B. Schölkopf and A. J. Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[107] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

[108] G. L. Sleijpen and H. A. Van der Vorst. A Jacobi–Davidson iteration method for linear eigenvalue problems. *SIAM review*, 42(2):267–293, 2000.

[109] D. C. Sorensen. Implicit application of polynomial filters in ak-step Arnoldi method. *Siam journal on matrix analysis and applications*, 13(1):357–385, 1992.

[110] D. C. Sorensen. Implicitly restarted Arnoldi/Lanczos methods for large scale eigenvalue calculations. In *Parallel Numerical Algorithms*, pages 119–165. Springer, 1997.

[111] D. A. Spielman and N. Srivastava. Graph sparsification by effective resistances. *SIAM Journal on Computing*, 40(6):1913–1926, 2011.

[112] 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*, pages 81–90. ACM, 2004.

[113] D. A. Spielman and S.-H. Teng. Spectral sparsification of graphs. *SIAM Journal on Computing*, 40(4):981–1025, 2011.

[114] D. A. Spielman and S.-H. Teng. A local clustering algorithm for massive graphs and its application to nearly linear time graph partitioning. *SIAM Journal on Computing*, 42(1):1–26, 2013.

[115] D. A. Spielman and S.-H. Teng. Nearly linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems. *SIAM Journal on Matrix Analysis and Applications*, 35(3):835–885, 2014.

[116] E. M. Stein. Interpolation of linear operators. *Transactions of the American Mathematical Society*, 83(2):482–492, 1956.

[117] G. Stewart. Accelerating the orthogonal iteration for the eigenvectors of a Hermitian matrix. *Numerische Mathematik*, 13(4):362–376, 1969.

[118] K. Stüben. A review of algebraic multigrid. *Journal of Computational and Applied Mathematics*, 128(1):281–309, 2001.

[119] D. Sutter, M. Berta, and M. Tomamichel. Multivariate trace inequalities. *Communications in Mathematical Physics*, 352(1):37–58, 2017.

[120] J. A. Tropp. Freedman's inequality for matrix martingales. *Electronic Communications in Probability*, 16:262–270, 2011.

[121] J. A. Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of Computational Mathematics*, 12(4):389–434, Aug 2012.

[122] J. A. Tropp. An introduction to matrix concentration inequalities. *Foundations and Trends® in Machine Learning*, 8(1–2):1–230, 2015.

[123] U. Trottenberg, C. W. Oosterlee, and A. Schuller. *Multigrid*. Academic press, 2000.

[124] A. Uhlmann. Relative entropy and the Wigner–Yanase–Dyson–Lieb concavity in an interpolation theory. *Communications in Mathematical Physics*, 54(1):21–32, 1977.

[125] P. Vaněk, J. Mandel, and M. Brezina. Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems. *Computing*, 56(3):179–196, 1996.

[126] A. Vershynina, E. A. Carlen, and E. H. Lieb. Matrix and operator trace inequalities. *Scholarpedia*, 8(4):30919, 2013.

[127] U. Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[128] G. H. Wannier. The structure of electronic excitation levels in insulating crystals. *Physical Review*, 52(3):191, 1937.

[129] P. Wesseling and C. W. Oosterlee. Geometric multigrid with applications to computational fluid dynamics. *Journal of Computational and Applied Mathematics*, 128(1):311–334, 2001.

[130] E. P. Wigner. Characteristic vectors of bordered matrices with infinite dimensions I. In *The Collected Works of Eugene Paul Wigner*, pages 524–540. Springer, 1993.

[131] E. P. Wigner and M. M. Yanase. Information contents of distributions. *Proceedings of the National Academy of Sciences of the United States of America*, 49(6):910, 1963.

[132] R. M. Wilcox. Exponential operators and parameter differentiation in quantum physics. *Journal of Mathematical Physics*, 8(4):962–982, 1967.

[133] H. Xie. A multigrid method for eigenvalue problem. *Journal of Computational Physics*, 274:550–561, 2014.

[134] H. Xie, L. Zhang, and H. Owhadi. Fast eigenpairs computation with operator adapted wavelets and hierarchical subspace correction. *SIAM Journal on Numerical Analysis*, 57(6):2519–2550, 2019.

[135] 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 transactions on pattern analysis and machine intelligence*, 29(1), 2007.

[136] H. Zhang. From Wigner–Yanase–Dyson conjecture to Carlen–Frank–Lieb conjecture. *Advances in Mathematics*, 365:107053, 2020.

[137] H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *Journal of computational and graphical statistics*, 15(2):265–286, 2006.