2019

Novel Computational Methods for Eigenvalue Problems

Ruihao Huang

NOVEL COMPUTATIONAL METHODS FOR EIGENVALUE PROBLEMS

By

Ruihao Huang

A DISSERTATION

Submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

In Mathematical Sciences

MICHIGAN TECHNOLOGICAL UNIVERSITY

2019

This dissertation has been approved in partial fulfillment of the requirements for the Degree of DOCTOR OF PHILOSOPHY in Mathematical Sciences.

Department of Mathematical Sciences

Dissertation Advisor:	*Dr. Jiguang Sun*
Committee Member:	*Dr. Chao Yang*
Committee Member:	*Dr. Lin Mu*
Committee Member:	*Dr. Zhengfu Xu*
Department Chair:	*Dr. Mark S. Gockenbach*

Contents

List of Figures

List of Tables

Preface

Several chapters in this dissertation are the result of collaborative work, and have been published in or submitted to referred journals.

Documentation of permission to reprint each article is documented in Appendix A.

- Chapter 2: *Recursive integral method for transmission eigenvalues.* This topic was supervised by Dr. Jiguang Sun[1]. It was previously published in Journal of Computational Physics.

- Chapter 3: *Recursive integral method with Cayley transformation.* This topic was supervised by Dr. Jiguang Sun. It was previously published in Numerical Linear Algebra with Applications.

- Chapter 4: *A Memory Efficient Multilevel Spectral Indicator Method.* This topic was supervised by Dr. Jiguang Sun. It was submitted.

- Chapter 5: *A new fast method of solving the high dimensional elliptic eigenvalue problem.* This is joint work with Dr. Lin Mu[2]. It was previously published in Applied Mathematics and Computation.

[1]Department of Mathematical Sciences, Michigan Technological University
[2]Department of Mathematics, University of Georgia

Acknowledgments

My sincere thanks go to my advisor, Dr. Jiguang Sun. During my graduate study, he has provided me with many opportunities. His dedication, suggestions and ideas have directed the course of my research. His encouragement has motivated me to find my passion in research, and I would not be where I am today without his support and guidance. I would like to express my sincere gratitude to my committee members: Dr. Lin Mu, Dr. Zhengfu Xu and Dr. Chao Yang for their effort. Particularly I owe thanks to Dr. Lin Mu, she was mentor when I was a research intern in Oak Ridge National Laboratory.

Many graduate students at MTU and friends deserve my gratitude. I would not be able to name all of them. But I must thank Dr. Sulin Wang, Dr. Chao Liang and Dr. Zhenchuan Wang.

I am incredibly grateful to my parents for their love, support and encouragement. Finally, I dedicate this dissertation to my girlfriend, Xing Ling. She has provided more support and love than I could have ever imaged, and kept me going through my toughest time.

Abstract

This dissertation focuses on novel computational method for eigenvalue problems.

In Chapter 1, preliminaries of functional analysis related to eigenvalue problems are presented. Some classical methods for matrix eigenvalue problems are discussed. Several PDE eigenvalue problems are covered. The chapter is concluded with a summary of the contributions.

In Chapter 2, a novel recursive contour integral method (**RIM**) for matrix eigenvalue problem is proposed. This method can effectively find all eigenvalues in a region on the complex plane with no a priori spectrum information. Regions that contain eigenvalues are subdivided and tested recursively until the size of region reaches specified precision. The method is robust, which is demonstrated using various examples.

In Chapter 3, we propose an improved version of **RIM** for non-Hermitian eigenvalue problems, called **SIM-M**. By incorporating Cayley transformation and Arnoldi's method, the main computation cost of solving linear systems is reduced significantly. The numerical experiments demonstrate that **RIM-M** gains significant speed-up over **RIM**.

In Chapter 4, we propose a multilevel spectral indicator method (**SIM-M**) to address the memory requirement for large sparse matrices. We modify the indicator of **RIM-M** such that it requires much less memory. Matrices from University of Florida Sparse Matrix Collection are tested, suggesting that a parallel version of **SIM-M** has the potential to be efficient.

In Chapter 5, we develop a novel method to solve the elliptic PDE eigenvalue problem. We construct a multi-wavelet basis with Riesz stability in $H_0^1(\Omega)$. By incorporating multi-grid discretization scheme and sparse grids, the method retains the optimal convergence rate for the smallest eigenvalue with much less computational cost.

Chapter 1

Introduction

Abstract

This chapter contains a brief introduction of the spectral theory for linear operators, eigenvalue problems of partial differential equations, matrix eigenvalue problems and applications. At the end of the chapter, the main contributions of the dissertation are discussed.

1.1 Functional Analysis

In this section, we present some fundamental results for spectral theory of linear operators [1].

Let X and Y be normed spaces. An operator $T : X \to Y$ is said to be linear if

$$T(\alpha x_1 + \beta x_2) = \alpha T x_1 + \beta T x_2 \quad \text{for all} \quad \alpha, \beta \in \mathbb{C}, x_1, x_2 \in X$$

and bounded if

$$\|Tx\|_Y \leq C \|x\|_X \quad \text{for all} \quad x \in X$$

for some constant C. Here $\|\cdot\|_Y$ and $\|\cdot\|_X$ are norms defined on X and Y, respectively. We say an operator T is continuous if, for every convergent sequence $\{x_n\}$ in X with limit x, we have

$$Tx_n \to Tx \quad \text{in } Y \quad \text{as} \quad n \to \infty$$

A linear operator is continuous if and only if it is bounded.

Definition 1.1.0.1. *We denote the set of all the continuous linear operators from X to Y by $\mathcal{L}(X,Y)$. Particularly, when $Y = X$, we write $\mathcal{L}(X)$. The set $\mathcal{L}(X,Y)$ is a linear space. The norm of a bounded linear operator $T : X \to Y$ is defined as*

$$\|T\|_{\mathcal{L}(X,Y)} = \sup_{x \neq 0, x \in X} \frac{\|Ax\|_Y}{\|x\|_X}$$

For simplicity, we use $\|T\|$ to denote $\|T\|_{\mathcal{L}(X,Y)}$.

Definition 1.1.0.2. *Let X and Y be normed spaces. A sequence of linear operators $\{T_n\}$ from X to Y is said to converge uniformly to a linear operator $T \in \mathcal{L}(X,Y)$ if*

$$\lim_{n \to \infty} \|T - T_n\| = 0.$$

Definition 1.1.0.3. *Let X be a normed space. A linear functional $f : X \to K$ is a linear operator such that $K = \mathbb{R}$ if X is a real vector space or $K = \mathbb{C}$ if X is a complex vector space. The set of all bounded linear functionals on X, denoted as X', is a normed space.*

Next we introduce the adjoint operator.

Definition 1.1.0.4. *Let X and Y be Hilbert spaces and $T : X \to Y$ be a bounded linear operator. The Hilbert adjoint operator T^* is defined as $T^* : Y \to X$ such that for all $x \in X$ and $y \in Y$*

$$(Tx, y)_Y = (x, T^*y)_X.$$

Definition 1.1.0.5. *A bounded linear operator $T : X \to X$ is said to be*

1. *self-adjoint or Hermitian if $T^* = T$,*

2. *unitary if T is bijective and $T^* = T^{-1}$,*

3. normal if $TT^* = T^*T$.

Let X be a complex normed space and $T : X \to X$ be a bounded linear operator. The following theorem gives the definition of the spectral radius of T.

Theorem 1.1.1. *Let* $T \in \mathcal{L}(X)$*. The limit*

$$r_\sigma(T) := \lim_{k \to \infty} \left\| T^k \right\|^{1/k}$$

exists and is called the spectral radius of T*.*

Let the operator be defined as

$$T_z = T - zI,$$

where $z \in \mathbb{C}$ and I is the identity operator. If T_z has an inverse, denoted by

$$R_z(T) = (T - zI)^{-1},$$

it is called the resolvent operator of T.

Definition 1.1.1.1. *Let* X *be a complex normed space and* $T : X \to X$ *a linear operator. A regular value* z *of* T *is complex number such that*

1. $R_z(T)$ *exist,*

2. $R_z(T)$ *is bounded, and*

3. $R_z(T)$ *is defined on a set which is dense in* X*.*

The resolvent set $\rho(T)$ of T is the set of all regular values z of T. Its complements $\sigma(T) := \mathbb{C} \backslash \rho(T)$ is called the spectrum of T. The spectrum $\sigma(T)$ can be partitioned into three disjoint set:

1. point spectrum $\sigma_p(T)$ is the set of z such that $R_z(T)$ does not exist. We call z the eigenvalue of T.

2. continuous spectrum $\sigma_c(T)$ is the set of z such that $R_z(T)$ exists and is defined on a dense set in X, but $R_z(T)$ is unbounded,

3. residual spectrum $\sigma_r(T)$ is the set of z such that $R_z(T)$ exists and the domain of $R_z(T)$ is not dense in X.

Definition 1.1.1.2. *Let $z \in \sigma_p(T)$ be an eigenvalue of some operator T. If*

$$T_z x : Tx - zx = 0 \tag{1.1.1}$$

for some $x \neq 0$, x is called an eigenfunction of T associated to z.

Let λ be an isolated eigenvalue of T such that there exists simple closed curves $\Gamma, \Gamma' \subset \rho(T)$ enclosing λ. Moreover, both Γ and Γ' do not include eigenvalues of T other than λ.

Next we give the definition of the spectrum projection which is main tool for the recursive integral method in Chapter 2.

$$P := \frac{1}{2\pi i}\int_\Gamma R(z)dz = \frac{1}{2\pi i}\int_\Gamma \frac{1}{T - zI}dz \tag{1.1.2}$$

To verify P defined above is a projection, we have

$$\begin{aligned} P^2 = \frac{1}{(2\pi i)^2}\int_\Gamma\int_{\Gamma'} R(z)R(z')dzdz' &= \frac{1}{(2\pi i)^2}\int_\Gamma\int_{\Gamma'} \frac{R(z) - R(z')}{z - z'}dzdz' \\ &= \frac{1}{2\pi i}\int_\Gamma R(z)dz \end{aligned}$$

In fact, P is the projection from X to the generalized eigenspace associated with λ when T is a compact operator. The eigenvalue problems we discuss in this thesis are related to compact operators, i.e., matrix eigenvalue problems from Chapter 2 to Chapter 4 and elliptic PDE eigenvalue problem for Chapter 5.

Definition 1.1.1.3. *Let X and Y be normed spaces. An operator $T : X \to Y$ is called a compact linear operator if T is linear and for every bounded subset M of X, $T(M)$ is relatively compact, i.e., $\overline{T(M)}$ is compact.*

Let $T : X \to X$ be a compact linear operator. The set of eigenvalues of T is at most countable and 0 is the only possible accumulation point. Every spectral values $\lambda \neq 0$ is an eigenvalue. If X is infinite dimensional, then $0 \in \sigma(T)$. Also for an eigenvalue $\lambda \neq 0$, the dimension of associated eigenspace of T is finite.

Next we define the Sobolev spaces. Let

$$W^{s,p}(\Omega) = \{f \in L^p(\Omega) | \partial^{\boldsymbol{\alpha}} f \in L^p \text{ for all } |\boldsymbol{\alpha}| \leq s\}$$

be the Sobolev space with associated norm

$$\|f\|_{W^{s,p}(\Omega)} = \left(\Sigma_{|\boldsymbol{\alpha}| \leq s} \int_\Omega |\partial^{\boldsymbol{\alpha} f(x)}|^p dx \right)^{1/p}.$$

When $p = 2$, we usually write

$$H^s(\Omega) = W^{s,2}(\Omega).$$

1.2 PDE Eigenvalue Problems

In this section, we introduce two PDE eigenvalue problems.

1.2.1 Laplace Eigenvalue Problem

Consider the following Laplace eigenvalue problem in Ω with Dirichlet boundary condition

$$-\Delta u = \lambda u, \text{ in } \Omega, \tag{1.2.3}$$

$$u = 0, \text{ on } \partial\Omega. \tag{1.2.4}$$

The variational formulation is to find $\lambda \in \mathbb{R}$ and non-trivial $u \in H_0^1(\Omega)$

$$a(u, v) := \int_\Omega \nabla u \cdot \nabla v dx = \lambda(u, v) \quad \text{for all } v \in H_0^1(\Omega). \tag{1.2.5}$$

We define the solution operator $T : L^2(\Omega) \to L^2(\Omega)$ which maps f to the solution u, i.e., $Tf = u$ and consequently,

$$a(Tf, v) = (f, v) \quad for \quad all \quad v \in H_0^1(\Omega).$$

Thus Laplace eigenvalue problem could rewrite as

$$\lambda(u, v) = a(\lambda Tu, v) = a(u, v) \quad \text{for all } v \in H_0^1(\Omega).$$

which is equivalent to the operator eigenvalue problem

$$\lambda Tu = u.$$

Thus λ is a Dirichlet eigenvalue of variational formulation if and only if $\frac{1}{\lambda}$ is an eigenvalue of operator T. In chapter 5, we will discuss more about Laplace eigenvalue problem and our novel method of solving the high dimensional elliptic eigenvalue problem.

1.2.2 Transmission Eigenvalue Problem

Let $D \subset \mathbb{R}^d, d = 2, 3$, be an open bounded domain with a Lipschitz boundary ∂D. Let k be the wave number of the incident plane wave $u^i = e^{ikx\cdot p}$, where $x, p \in \mathbb{R}^d$, $|p| = 1$. Denote the index of refraction by $n(x)$ such that $n(x) \geq n_0 > 1$. The direct scattering problem by the inhomogeneous medium D is to find the total field $u(x)$ satisfying

$$\Delta u + k^2 n(x) u = 0, \qquad \text{in } D, \tag{1.2.6a}$$

$$\Delta u + k^2 u = 0, \qquad \text{in } \mathbb{R}^d \setminus D, \tag{1.2.6b}$$

$$u(x) = e^{ikx\cdot p} + u^s(x), \qquad \text{in } \mathbb{R}^d, \tag{1.2.6c}$$

$$\lim_{r\to\infty} \sqrt{r}\left(\frac{\partial u^s}{\partial r} - iku^s\right) = 0, \tag{1.2.6d}$$

where u^s is the scattered field and $r = |x|$. The Sommerfeld radiation condition (1.2.6d) is assumed to hold uniformly with respect to $\hat{x} = x/|x|$.

The associated transmission eigenvalue problem is to find $\lambda := k^2 \in \mathbb{C}$ and non-trivial w and v such that

$$\Delta w + \lambda n(x) w = 0, \qquad \text{in } D, \tag{1.2.7a}$$

$$\Delta v + \lambda v = 0, \qquad \text{in } D, \tag{1.2.7b}$$

$$w - v = 0, \qquad \text{on } \partial D, \tag{1.2.7c}$$

$$\frac{\partial w}{\partial \nu} - \frac{\partial v}{\partial \nu} = 0, \qquad \text{on } \partial D, \tag{1.2.7d}$$

where ν is the unit outward normal to ∂D.

We first transform (1.2.7) into a fourth order problem. Let $z = v - w \in H_0^2(D)$. Subtracting (1.2.7a) from (1.2.7b), we have

$$(\triangle + \lambda n(x)) z = -\lambda (n(x) - 1) v,$$

which implies

$$(n(x) - 1)^{-1} (\triangle + \lambda n(x)) z = -\lambda v.$$

Applying $(\triangle + \lambda)$ to the above equation, (1.2.7b) leads to

$$(\triangle + \lambda) \frac{1}{n(x) - 1} (\triangle + \lambda n(x)) z = 0. \tag{1.2.8}$$

To obtain a mixed formulation, let $y = \frac{1}{n(x)-1}(\triangle + \lambda n(x)) z$. Hence

$$\begin{aligned} (\Delta + \lambda) y &= 0, \\ \frac{1}{n(x) - 1} (\Delta + \lambda n(x)) z &= y. \end{aligned}$$

The associated weak problem is to find $(\lambda, z, y) \in \mathbb{C} \times H_0^1(D) \times H^1(D)$ such that

$$\begin{aligned} (\nabla y, \nabla \phi) &= \lambda (y, \phi) \quad \text{for all } \phi \in H_0^1(D), \\ (\nabla z, \nabla \varphi) + ((n(x) - 1) y, \varphi) &= \lambda (n(x) z, \varphi) \quad \text{for all } \varphi \in H^1(D). \end{aligned}$$

In the following, we describe a simple mixed finite element method proposed in [2]. Let a triangular mesh for $D \subset \mathbb{R}^2$ or a tetrahedral mesh for $D \subset \mathbb{R}^3$ be given. Define the linear Lagrange finite element spaces

$$V_h = \text{the space of the linear Lagrange elements on } D,$$

$$\begin{aligned} V_h^0 &= V_h \cap H_0^1(D) \\ &= \text{the subspace of functions in } V_h \text{ with vanishing DoF on } \partial D, \\ V_h^{\mathcal{B}} &= \text{the subspace of functions in } V_h \text{ with vanishing DoF in } D, \end{aligned}$$

where DoF stands for degrees of freedom. The discrete weak formulation is to find $(\lambda_h, z_h, y_h) \in \mathbb{C} \times V_h^0 \times V_h$ such that

$$\begin{aligned} (\nabla y_h, \nabla \phi_h) &= \lambda_h (y_h, \phi_h) \quad \text{for all } \phi_h \in V_h^0, \\ (\nabla z_h, \nabla \varphi_h) + ((n(x)-1)y_h, \varphi_h) &= \lambda_h (n(x) z_h, \varphi_h) \quad \text{for all } \varphi_h \in V_h. \end{aligned}$$

Let $\psi_1, \ldots, \psi_K$ be a basis for V_h^0 and $\psi_1, \ldots, \psi_K, \psi_{K+1}, \ldots, \psi_T$ be a basis for V_h such that $z_h = \sum_{i=1}^K z_i \psi_i$ and $y_h = \sum_{i=1}^T y_i \psi_i$. Let $\boldsymbol{z} = (z_1, \ldots, z_K)'$ and $\boldsymbol{y} = (y_1, \ldots, y_T)'$, where $'$ denotes the transpose. The matrix problem is

$$\begin{aligned} S_{K\times T}\boldsymbol{y} &= \lambda_h M_{K\times T}\boldsymbol{y}, \\ S_{T\times K}\boldsymbol{z} + M_{T\times T}^{n(x)-1}\boldsymbol{y} &= \lambda_h M_{T\times K}^{n(x)}\boldsymbol{z}, \end{aligned}$$

where

$$\begin{aligned} (S_{K\times T})_{i,j} &= (\nabla\psi_i, \nabla\psi_j), \quad 1 \le i \le K, 1 \le j \le T, \\ (S_{K\times T})_{i,j} &= (\nabla\psi_i, \nabla\psi_j), \quad 1 \le i \le T, 1 \le j \le K, \\ (M_{K\times T})_{i,j} &= (\psi_i, \psi_j), \quad 1 \le i \le K, 1 \le j \le T, \\ (M_{T\times K}^{n(x)})_{i,j} &= (n(x)\psi_i, \psi_j), \quad 1 \le i \le T, 1 \le j \le K, \\ (M_{T\times T}^{n(x)-1})_{i,j} &= ((n(x)-1)\psi_i, \psi_j), \quad 1 \le i \le T, 1 \le j \le T. \end{aligned}$$

The generalized eigenvalue problem is

$$\begin{pmatrix} S_{K\times T} & 0_{K\times K} \\ M_{T\times T}^{n(x)-1} & S_{T\times K} \end{pmatrix} \begin{pmatrix} \boldsymbol{y} \\ \boldsymbol{z} \end{pmatrix} = \lambda_h \begin{pmatrix} M_{K\times T} & 0_{K\times K} \\ 0_{T\times T} & M_{T\times K}^{n(x)} \end{pmatrix} \begin{pmatrix} \boldsymbol{y} \\ \boldsymbol{z} \end{pmatrix}.$$

For simplicity, we use λ instead of λ_h and write the above problem as

$$A\boldsymbol{x} = \lambda B\boldsymbol{x} \tag{1.2.9}$$

with

$$A = \begin{pmatrix} S_{K\times T} & 0_{K\times K} \\ M_{T\times T}^{n(x)-1} & S_{T\times K} \end{pmatrix}, \quad B = \begin{pmatrix} M_{K\times T} & 0_{K\times K} \\ 0_{T\times T} & M_{T\times K}^{n(x)} \end{pmatrix} \begin{pmatrix} \boldsymbol{y} \\ \boldsymbol{z} \end{pmatrix}, \quad \boldsymbol{x} = \begin{pmatrix} \boldsymbol{y} \\ \boldsymbol{z} \end{pmatrix}.$$

Note that (1.2.9) is non-Hermitian. In general, there exist complex eigenvalues. In chapter 2, we will introduce a novel method of solving (1.2.9) based on spectral projection (1.1.2).

1.3 Matrix Eigenvalue Problems

In this section, we discuss classical methods of solving matrix eigenvalue problem [3] related the thesis.

Definition 1.3.0.1. *A complex number λ is called an eigenvalue of matrix A if there exists a nonzero vector x such that*

$$Ax = \lambda x$$

The vector x is called an eigenvector associated with λ.

Definition 1.3.0.2. *A complex number λ is called a generalized eigenvalue for the generalized eigenvalue problem*

$$Ax = \lambda Bx,$$

where $A, B \in \mathbb{C}^{n\times n}$ and B can be singular.

1.3.1 Krylov Subspaces

An important class of techniques knows as Krylov subspace methods extracts approximations from the following subspace

$$\mathcal{K}_m = \mathrm{Span}\{\boldsymbol{v}, A\boldsymbol{v}, A^2\boldsymbol{v}, \ldots, A^{k-1}\boldsymbol{v}\}$$

The Arnoldi's method tries to seek best approximation in $\mathcal{K}_m$ by orthogonal projection onto $\mathcal{K}_m$ for general non-Hermitian matrices. Let W be a matrix whose columns form an orthonormal basis for $\mathcal{K}_m$. One only to solve the reduced eigenvalue problem as following of size k

$$\hat{A} = W^H A W$$

When comes to generalized eigenvalue problem $Ax = \lambda Bx$, if B is non-singular, we could rewrite as $B^{-1}Ax = \lambda x$ then apply the Krylov subspace method. However when B is singular, the above method fails. Fortunately, we could fix it by Cayley transformation. We will discuss more details in Chapter 3.

1.3.2 Integral Based Eigensolvers

There are needs for computing eigenvalues of a nonlinear and/or non-Hermitian eigenvalue problem that lie in a given region in the complex plane. Also the convergence behaviors of Krylov subspace are rather complex for non-hermitian cases. Recently integral based method has become popular, it is a hybrid method of non-linear filtering (contour integrals of the resolvent) and subspace iteration.

The original problem considered by Polizzi [4] .i.e FEAST is as follows

$$Ax = \lambda Bx,$$

A is n by n Hermitian and B is n by n positive definite matrix. The goal is to compute all the eigenvalues and the associated eigenvectors in the specified interval (a, b). For simplicity we assume a and b are not the generalized eigenvalues We define the spectral projection for the generalized eigenvalue problem,

$$P = \frac{1}{2\pi i} \int_{\Gamma} (zB - A)^{-1} dz. \tag{1.3.10}$$

Here Γ is a circle centered at $(a+b)/2$ with radius $r = \frac{b-a}{2}$. Assume that there are only $k \ll n$ eigenvalues inside Γ. Let V_k be a matrix whose columns are k linear independent random vectors and $Q = PV_k$. Then the original problem $Ax = \lambda Bx$ reduces to a generalized eigenvalue problem of size k

$$Q^T AQ\Phi = \lambda Q^T BQ\Phi,$$

where Q can only be computed by numerical quadratures.

Simple version of FEAST Algorithm:

1. Select $k_0 > k$ random matrix $V_{n\times k_0}$.
2. Set $Q = 0$ with $Q \in \mathbb{R}^{n\times k_0}$ and $r = \frac{b-a}{2}$
3. for $j = 1, \ldots, N_e$

    compute θ_j = -$\pi/2(x_j - 1)$ and $z_j = \frac{a+b}{2} + re^{i\theta_j}$

    compute $(z_j B - A)Q_j = V$

    compute $Q = Q - (w_j/2)\mathcal{R}[re^{i\theta_j} Q_j]$

4. solve $Q^T AQ\Phi = \lambda Q^T BQ\Phi$ to obtain k_0 eigenvalues and eigenvectors $\Phi_{k_0\times k_0}$
5. compute $X_{n\times k_0} = Q\Phi_{k_0\times k_0}$
6. check convergence for the trace of the eigenvalues. If refinement is needed, compute $V = BX$ and go to step 2.

Here (x_j, w_j) are any quadrature points.

1.4 Application of Matrix Eigenvalue Problems in Data Mining

Singular value decomposition (SVD) is a factorization of a real or complex matrix A and it is a classical technique covered in almost linear algebra book. SVD has profound impact in data mining area, e.g. Principal Component Analysis (PCA).

PCA uses an orthogonal transformation to re-combine a set of observations such that the first principal component has the largest possible variance, which accounts for as much variability of data as possible. PCA has been used to reduce the redundancy of data as pre-processing step thus make predictive models much more robust. The famous Netflix Prize problem is to predict how well its users might like individual movies, so that it could recommend movies to them. The problem could be modeled as a matrix completion problem and a low rank SVD approximation is winner solution [5].

Another important application is the PageRank algorithm to measure the importance of website pages. The most significant step in PageRank is power method for computing the associated eigenvector of the largest eigenvalue [6].

1.5 Main Contributions

In chapter 2, we propose the recursive integral method (RIM) for transmission eigenvalues since the discrete problem (1.2.9) leads to non-Hermitian matrix eigenvalue problem with very complicated spectrum and only a small portion of eigenvalues are needed. By splitting the area of interest based on the spectrum projections until it reaches the tolerance, our method is robust and suitable for parallel computation. In chapter 3, we optimize the RIM algorithm by introducing Arnolid's method with Cayley transformation. Thus the RIM-C achieves comparable efficiency as 'eigs' in Matlab. Finally in chapter 4, we propose a new indicator function with much less memory requirement and test it using several examples in data science. This new version of spectral indicator method (SIM-M) shows great potential in large matrix eigenvalue computation. Chapter 5 contains novel method which aims to solve high dimensional elliptic eigenvalue problem efficiently in tensorized domain. We combine the multi-wavelet basis with multi-grid method to compute the smallest eigenvalue for high dimensional elliptic eigenvalue problem. The condition number of the resulting matrice does not change with mesh size and the dimension of the problem.

Chapter 2

Recursive Integral Method for Transmission Eigenvalues[1]

Abstract

Transmission eigenvalue problems arise from inverse scattering theory for inhomogeneous media. These non-selfadjoint problems are numerically challenging because of a complicated spectrum. In this chapter, we propose a novel recursive contour integral method for matrix eigenvalue problems from finite element discretizations of transmission eigenvalue problems. The technique tests (using an approximate spectral projection) if a region contains eigenvalues. Regions that contain eigenvalues are subdivided and tested recursively until eigenvalues are isolated with a specified precision. The method is fully parallel and requires no a priori spectral information. Numerical examples show the method is effective and robust.

[1]This chapter has been published as an article in Journal of Computational Physics. https://doi.org/10.1016/j.jcp.2016.10.001

2.1 Introduction

The transmission eigenvalue problem [7, 8, 9, 10] has important applications in the inverse scattering theory for inhomogeneous media. It is nonlinear and non-selfadjoint. Early study focused on showing that transmission eigenvalues form at most a discrete set since sampling methods for reconstructing the support of an inhomogeneous medium fail if the interrogating frequency corresponds to a transmission eigenvalue [10]. Later, it was realized that transmission eigenvalues can be obtained from the scattering data and used to reconstruct the physical properties of the unknown target [8].

Recently, significant efforts have been devoted to develop numerical methods for transmission eigenvalues [11, 9, 2, 12, 13, 14, 15, 16, 17, 18, 1]. In [11], Colton et al. proposed three finite element methods. A mixed method based on a fourth order formulation was developed in [2]. An and Shen [14] proposed an efficient spectral-element method for two-dimensional radially-stratified media. A conforming finite element method was introduced by Sun in [9], where real transmission eigenvalues are computed as roots of a nonlinear function whose values are generalized eigenvalues of a related fourth order problem. Using a fourth order formulation, Cakoni et al. [16] proposed a new mixed finite element method and proved convergence based on Osborn's theory [19]. Li et al. [17] developed a finite element method by considering a quadratic eigenvalue problem. Integral equations are used to compute transmission eigenvalues as well. In [20], Cossonniére and Haddar formulated the transmission eigenvalue problem as a nonlinear integral eigenvalue problem. The same formulation was used by Kleefeld in [15]. To solve the nonlinear eigenvalue problem, Kleefeld adopted the method proposed by Beyn [21] using spectrum projection. Some non-traditional methods, including the linear sampling method [22] and the inside-out duality [23], were proposed to search for eigenvalues using scattering data. We also refer the readers to other methods in [24, 25, 26, 27, 12] for the transmission eigenvalue problem and the related source problem.

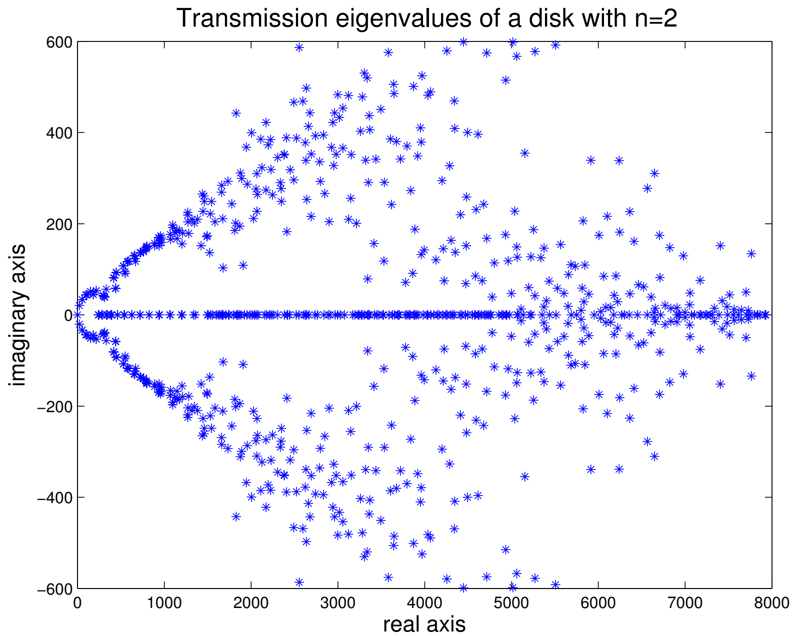


Figure 2.1: Transmission eigenvalues on the complex plane: a disk with radius 1/2 and index of refraction $n = 2$.

Since the transmission eigenvalue problem is non-selfadjoint, finite element discretizations usually lead to non-Hermitian matrix eigenvalue problems. In addition, the spectrum is very complicated in general (see Fig. 2.1). These characteristics suggest that most existing eigenvalue solvers might not be suitable for transmission eigenvalues.

In this chapter, we propose a novel recursive integral method (**RIM**) to compute generalized matrix eigenvalues resulting from finite element discretizations of the transmission eigenvalue problem. We aim at developing an eigensolver for problems with the following features:

1) the problem is non-selfadjoint,

2) spectrum is complicated,

3) no a priori information, such as number of eigenvalues inside the given region,

is available,

4) interior eigenvalues are needed.

Spectrum projection using contour integrals on the complex plane is a classical approach in the operator spectral theory [28]. Recently, contour integral type methods become popular [29, 4, 21, 30] (see also [31]). These methods use Cauchy integrals of the resolvent to compute spectrum projections onto the generalized eigenspace corresponding to the eigenvalues inside a simple closed curve on the complex plane [32]. The original problem is then reduced to a subspace problem.

In contrast, **RIM** tests a region on the complex plane using spectrum projections. An indicator is calculated to decide if the region contains eigenvalue(s) or not. In the case of a positive answer, the region is subdivided and tested for eigenvalues recursively. **RIM** does not actually compute eigenvalues of a subspace problem. The eigenvalues are obtained using a series of domain decompositions, which is the major difference from the existing integral methods.

The rest of the chapter is arranged as follows. In Section 2.2 we present **RIM**. We discuss some implementation details in Section 2.3. Section 2.4 contains a comprehensive numerical study. We conclude in Section 2.5 with some discussions.

2.2 A Recursive Contour Integral Method

In Chapter 1.3.2, we introduced Transmission Eigenvaule problem and in this section, we propose a novel eigensolver for (1.2.9) using spectrum projections. We start with some classical results of the operator spectral theory (see, e.g., [28]). Let $T : \mathcal{X} \to \mathcal{X}$ be a bounded linear operator on a complex Hilbert space $\mathcal{X}$. The resolvent set of T is defined as

$$\rho(T) = \{z \in \mathbb{C} : (z - T)^{-1} \text{ exists as a bounded operator on } \mathcal{X}\}. \qquad (2.2.1)$$

For any $z \in \rho(T)$,

$$R_z(T) = (z - T)^{-1} \tag{2.2.2}$$

is the resolvent operator of T. The spectrum of T is $\sigma(T) = \mathbb{C} \setminus \rho(T)$. We assume that T has only point spectrum, i.e., each $\lambda \in \sigma(T)$ is an isolated eigenvalue of T. In addition, we assume that the eigenspace associated with λ is finite dimensional. Let α be the least positive integer such that

$$N\left((\lambda - T)^{\alpha}\right) = N\left((\lambda - T)^{\alpha+1}\right),$$

where N denotes the null space. Then $m = \dim N\left((\lambda - T)^{\alpha}\right)$ is called the algebraic multiplicity of λ. The functions in $N\left((\lambda - T)^{\alpha}\right)$ are called the generalized eigenfunctions of T corresponding to λ. Note that the geometric multiplicity of λ is defined as $\dim N(\lambda - T)$.

Let Γ be a simple closed curve on the complex plane $\mathbb{C}$ lying in $\rho(T)$ which contains m eigenvalues, counting multiplicity, of T: $\lambda_j, j = 1, \ldots, m$. Define

$$P = \frac{1}{2\pi i} \int_{\Gamma} R_z(T) dz.$$

It is well-known that P is a projection onto the space of generalized eigenfunctions $\boldsymbol{u}_j$ associated with $\lambda_j, j = 1, \ldots, m$. The projection P depends only on eigenvalues inside Γ and is called the spectrum projection [28].

The following is the main idea behind **RIM**. Let $\boldsymbol{f} \in \mathcal{X}$ be a random element. If there are no eigenvalues inside Γ, $P\boldsymbol{f} = \mathbf{0}$. Otherwise, if there are m eigenvalues $\lambda_j, j = 1, \ldots, m$, $P\boldsymbol{f} \neq \mathbf{0}$ provided that $\boldsymbol{f}$ has components in $\boldsymbol{u}_j, j = 1, \ldots, m$. Thus $P\boldsymbol{f}$ can be used to decide if a region contains eigenvalues of T or not. If a region contains eigenvalue(s), it is partitioned into smaller regions. Then one computes $P\boldsymbol{f}$ for these small regions. The process is repeated until the size of the region is smaller than a given precision.

Our goal is to find all the eigenvalues of T in the interior of Γ, denoted by S. Let $\{z_j, \omega_j, j = 1, \ldots, W\}$ be a quadrature rule, where z_j's are the quadrature points on

Γ and ω_j's are the associated weights. We approximate the projection $P\boldsymbol{f}$ by

$$P\boldsymbol{f} \approx \frac{1}{2\pi i}\sum_{j=1}^{W}\omega_j R_{z_j}(T)\boldsymbol{f}. \tag{2.2.3}$$

Let $\boldsymbol{x}_j, j = 1, \ldots, W$, be such that

$$(z_j - T)\boldsymbol{x}_j = \boldsymbol{f}, \quad j = 1, \ldots, W.$$

Then we have

$$P\boldsymbol{f} = \frac{1}{2\pi i}\sum_{j=1}^{W}\omega_j \boldsymbol{x}_j. \tag{2.2.4}$$

According to the above discussion, $\|P\boldsymbol{f}\|_{\mathcal{X}}$ can be used to decide if there are eigenvalues in S, i.e.,

(i) if $\|P\boldsymbol{f}\|_{\mathcal{X}} \neq 0$, there exists at least one eigenvalue in S;

(ii) if $\|P\boldsymbol{f}\|_{\mathcal{X}} = 0$, there is no eigenvalue in S.

In Case (i), we divide S into subregions and recursively repeat this procedure. The process terminates when the size of the region $h(S)$ is smaller than the given precision ϵ.

The algorithm of **RIM** is as follows.

RIM$(S, \epsilon, \boldsymbol{f})$

Input: a region S, precision ϵ, a randomly chosen $\boldsymbol{f}$

Output: λ, eigenvalue(s) of T in S

1. Approximate $P\boldsymbol{f}$ by (2.2.4);
2. Decide if S contains eigenvalue(s) using $\|P\boldsymbol{f}\|_{\mathcal{X}}$:
   - No. exit.
   - Yes. compute the size $h(S)$ of S,

- if $h(S) > \epsilon$, partition S into subregions $S_j, j = 1, \dots N$.

  for $j = 1$ to N

  RIM$(S_j, \epsilon, \boldsymbol{f})$

  end

- if $h(S) \leq \epsilon$, output the eigenvalues and exit.

In practice, we do need a threshold δ_0 to distinguish between $\|P\boldsymbol{f}\|_{\mathcal{X}} \neq 0$ and $\|P\boldsymbol{f}\|_{\mathcal{X}} = 0$. We postpone the discussion to the next section.

Note that the finite element discretization in Section 1.2.2 leads to a generalized matrix eigenvalue problem (1.2.9). The corresponding resolvent is defined as

$$R_z(A, B) = (zB - A)^{-1} \tag{2.2.5}$$

for z in the resolvent set of the matrix pencil (A, B). The spectrum projection onto the generalized eigenspace corresponding to eigenvalues enclosed by Γ is

$$P(A, B) = \frac{1}{2\pi i} \int_\Gamma (zB - A)^{-1} dz. \tag{2.2.6}$$

For any vector $\boldsymbol{f} \in \mathbb{C}^n$, we need to compute

$$\begin{aligned} P\boldsymbol{f} &= \frac{1}{2\pi i} \int_\Gamma R_z(A, B)\boldsymbol{f} dz \\ &\approx \frac{1}{2\pi i} \sum_{j=1}^{W} \omega_j R_{z_j}(A, B)\boldsymbol{f} \\ &= \frac{1}{2\pi i} \sum_{j=1}^{W} \omega_j \boldsymbol{x}_j, \end{aligned} \tag{2.2.7}$$

where $\boldsymbol{x}_j$'s are the solutions of the following linear systems

$$(z_j B - A)\boldsymbol{x}_j = \boldsymbol{f}, \quad j = 1, \dots, W. \tag{2.2.8}$$

If there are no eigenvalues inside Γ, then $P = 0$ and thus $P\boldsymbol{f} = \boldsymbol{0}$ for all $\boldsymbol{f} \in \mathbb{C}^n$.

2.3 Implementation

In this section, we discuss the implementation of the matrix version of **RIM**. We choose the search region S to be a rectangle on the complex plane. In particular, we assume that the width and length are of similar sizes. Otherwise, one can pre-divide S into smaller rectangles. We call S *admissible* if the indicator $\delta_S := |P_S \boldsymbol{f}| > \delta_0$, where δ_0 is the threshold value we shall specify later. We divide an admissible rectangle S into non-overlapping sub-rectangles and compute the indicators until the regions are smaller than the given precision $\epsilon > 0$.

There are several key points in the implementation:

(1) a suitable quadrature rule for (2.2.3),

(2) a mechanism to solve (2.2.8),

(3) a suitable threshold δ_0.

For (1), we use the midpoint of each edge of S as the quadrature point and four points in total. It is every coarse. However, the numerical examples show that it is enough. In fact, the indicator does not need to be computed exactly. Note that other contour integral methods use more quadrature points. For example, twenty-five quadrature points are used in [21].

For (2), to solve the linear systems (2.2.8), MATLAB "\" is used in the current implementation. It is efficient for systems of tens of thousands unknowns in MATLAB. Note that other iterative solvers, such as "lsqr" in Matlab also works.

According to the algorithm in the previous section, we first pick up a random vector $\boldsymbol{f}$ and calculate $P\boldsymbol{f}$. If the norm of $P\boldsymbol{f}$ is zero, there is no eigenvalue inside the region. Otherwise, there are eigenvalues inside. In practice, the norm of $P\boldsymbol{f}$ is never zeros due to quadratures, linear solvers, and machine precision. Consequently, for (3), we need to choose a suitable value δ_0. We denote by $\mathcal{R}(P)$ the range of P, which coincides with the finite dimensional generalized eigenspace associated with the eigenvalues inside Γ. Let $\boldsymbol{\phi}_j, j = 1, \ldots, M$, be an orthonormal basis of $\mathcal{R}(P)$. Let $\boldsymbol{f}$ be a randomly chosen vector and

$$P\boldsymbol{f} = \boldsymbol{f}|_{\mathcal{R}(P)} = \sum_{j=1}^{M} a_j \boldsymbol{\phi}_j, \tag{2.3.9}$$

where $a_j = (\boldsymbol{f}, \boldsymbol{\phi}_j)$. To decide if a region contains eigenvalues, the following two elements need to be considered:

(i) $|P\boldsymbol{f}|$ can be relatively small when there is an eigenvalue(s) in S.

(ii) $|P\boldsymbol{f}|$ can be relatively large when there is no eigenvalue in S.

Case (i) can happen if $\left|\boldsymbol{f}|_{\mathcal{R}(P)}\right|$ is small, i.e., $\sum_{j=1}^{M} a_i^2$ is small. Our solution is to normalized $P\boldsymbol{f}$ and project once again. The indicator is set to be

$$\delta_S = \left| P\left(\frac{P\boldsymbol{f}}{|P\boldsymbol{f}|}\right)\right|. \tag{2.3.10}$$

Remark 2.3.0.1. Analytically, $P^2\boldsymbol{f} = P\boldsymbol{f}$. Numerically, they are not the same. In particular, we approximate the spectrum projection using just four quadrature points.

Case (ii) happens if there exists eigenvalue(s) lies outside S but close to it. In fact, this must happen when **RIM** zooms into the neighborhood of an eigenvalue. Fortunately, **RIM** has an interesting *self-correction* property. This property will be illustrated in the next section.

Here are some details of the implementation:

1. A rectangular search region S.

2. Matlab "\" for the linear systems.

3. One quadrature point for each edge of S.

4. One random vector $\boldsymbol{f}$.

5. Projections are computed twice using (2.3.10).

6. $\delta_0 = 1/10$, i.e., if $\delta_S > 1/10$, S is admissible.

The matrix version of **RIM** is as follows.

M-RIM$(A, B, S, \epsilon, \delta_0, \boldsymbol{f})$

Input: matrices A, B, region S, precision ϵ, thresh hold δ_0, random vector $\boldsymbol{f}$.

Output: generalized eigenvalue(s) λ inside S

1. Compute δ_S using (2.3.10).
2. Decide if S contains eigenvalue(s).
   - If $\delta_S < \delta_0$. Exit.
   - Otherwise, compute the size $h(S)$ of S.
     - if $h(S) > \epsilon$, partition S into subregions $S_j, j = 1, \ldots N$.

       for $j = 1$ to N

       M-RIM$(A, B, S_j, \epsilon, \delta_0, \boldsymbol{f})$.

       end
     - if $h(S) \le \epsilon$,

       set λ to be the center of S.

       output λ and exit.

2.4 Numerical Examples

We present some examples to show the performance of **RIM**. All the computation is done using Matlab on a Macbook Pro with a 3G Hz Intel Core i7 and 16GB 1600 MHz DDR3 memory.

Remark 2.4.0.1. By "exact eigenvalues", we mean the generalized eigenvalues of (1.2.9), which are the finite element approximations of the transmission eigenvalues. These "exact" generalized eigenvalues are obtained by "eigs" in MATLAB.

2.4.1 Effectiveness

Example 1: We consider a disc D with radius 1/2 and the index of refraction $n(x) = 16$. A triangular mesh with $h \approx 0.05$ is used to generate two 1018×1018 matrices A and B. We consider a search region $S = [3, 9] \times [-3, 3]$. The exact eigenvalues in S are

$$\lambda_1 = 3.994539, \quad \lambda_2 = 6.935054, \quad \lambda_3 = 6.939719.$$

With $\epsilon = 10^{-3}$, **RIM** successfully returns 3 eigenvalues

$$\lambda_1^{\mathbf{RIM}} = (3.994629 \pm 10^{-3}) \pm 10^{-3}i,$$
$$\lambda_2^{\mathbf{RIM}} = (6.935059 \pm 10^{-3}) \pm 10^{-3}i,$$
$$\lambda_3^{\mathbf{RIM}} = (6.939941 \pm 10^{-3}) \pm 10^{-3}i,$$

where $i = \sqrt{-1}$.

As the second search region, we choose $S = [22, 25] \times [-8, 8]$. Two exact eigenvalues in S are

$$\lambda_1 = 24.158567 + 5.690114i, \quad \lambda_2 = 24.158567 - 5.690114i.$$

RIM outputs the following

$$\lambda_1^{\mathbf{RIM}} = (24.158813 \pm 10^{-3}) - (5.690308 \pm 10^{-3})i,$$
$$\lambda_2^{\mathbf{RIM}} = (24.158813 \pm 10^{-3}) + (5.690063 \pm 10^{-3})i.$$

In Fig. 2.2, we show how **RIM** explores the region $S = [0, 30] \times [-6, 6]$. There are 16 eigenvalues in S including two complex ones. **RIM** finds all of them successfully.

Example 2: Let D be the unit square and $n = 16$. The matrices A and B are 1298×1298. The first search region is $S = [6, 9] \times [-1, 1]$. The exact eigenvalues are

$$\lambda_1 = 6.049528, \quad \lambda_2 = 6.051180, \quad \lambda_3 = 8.368568.$$

RIM gives the following eigenvalues

$$\lambda_1^{\mathbf{RIM}} = (6.049316 \pm 10^{-3}) \pm 10^{-3}i,$$

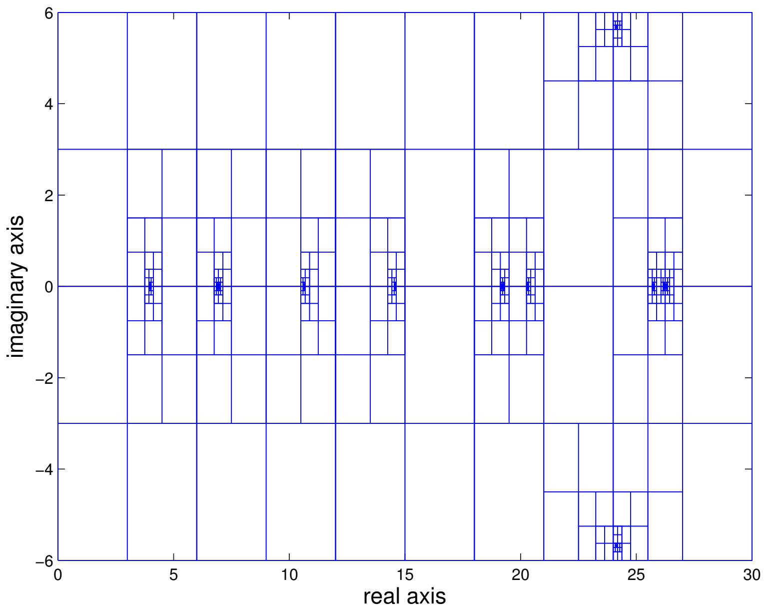


Figure 2.2: The regions explored by **RIM** with $S = [0, 30] \times [-6, 6]$ for **Example 1**. There are 16 eigenvalues. Some of them are clustered.

$$\lambda_2^{\mathbf{RIM}} = (6.051270 \pm 10^{-3}) \pm 10^{-3} i,$$

$$\lambda_3^{\mathbf{RIM}} = (8.368652 \pm 10^{-3}) \pm 10^{-3} i.$$

The second search region is $S = [20, 21] \times [-6, 6]$. The exact eigenvalues are

$$\lambda_1 = 20.573786 + 5.127225i,$$

$$\lambda_2 = 20.573786 - 5.127225i.$$

The eigenvalues computed by **RIM** are

$$\lambda_1^{\mathbf{RIM}} = (20.573730 \pm 10^{-3}) - (5.127441i \pm 10^{-3})i,$$

$$\lambda_2^{\mathbf{RIM}} = (20.573730 \pm 10^{-3}) + (5.126465i \pm 10^{-3})i.$$

Example 3: Let D be the L-shaped domain defined by

$$(-1, 1) \times (-1, 1) \setminus [0, 1] \times [-1, 0]$$

and $n = 16$. The matrices A and B are 978×978. The search region is $S = [2,3] \times [-1/2, 1/2]$. The exact eigenvalues are

$$\lambda_1 = 2.210247, \quad \lambda_2 = 2.50668, \quad \lambda_3 = 2.979671.$$

RIM computes the following eigenvalues

$$\begin{aligned}
\lambda_1^{\mathbf{RIM}} &= (2.210236 \pm 10^{-3}) \pm 10^{-3} i, \\
\lambda_2^{\mathbf{RIM}} &= (2.506683 \pm 10^{-3}) \pm 10^{-3} i, \\
\lambda_3^{\mathbf{RIM}} &= (2.979706 \pm 10^{-3}) \pm 10^{-3} i.
\end{aligned}$$

Example 4: We consider a 3D problem. Let D be the unit ball with the index of refraction $n = 4$. A tetrahedral mesh with the mesh size $h \approx 0.05$ is given. The mixed finite element method using the linear Lagrange elements leads to a 42606×42606 generalized matrix eigenvalue problem. Let $S = [10, 11] \times [-1/2, 1/2]$. There are three exact eigenvalues in S:

$$\lambda_1 = 10.345551, \quad \lambda_2 = 10.357927, \quad \lambda_3 = 10.369776.$$

RIM computes the following

$$\begin{aligned}
\lambda_1^{\mathbf{RIM}} &= (10.346875 \pm 10^{-3}) \pm 10^{-3} i, \\
\lambda_2^{\mathbf{RIM}} &= (10.353125 \pm 10^{-3}) \pm 10^{-3} i, \\
\lambda_3^{\mathbf{RIM}} &= (10.371875 \pm 10^{-3}) \pm 10^{-3} i.
\end{aligned}$$

These values are consistent with those given on Page 4 of [15]. Note that we actually compute the square of the transmission eigenvalues, i.e., $\kappa^2_{1,\mathbb{S}^2,4} \approx 9.8696$.

Example 5: Let D be the unit cube with the index of refraction $n = 16$. Again we generate a tetrahedral mesh with $h \approx 0.05$. The generalized eigenvalue problem is 46735×46735. Let $S = [4, 5] \times [-1/2, 1/2]$. The eigenvalue in S is

$$\lambda_1 = 4.328288.$$

RIM outputs

$$\lambda_1^{\mathbf{RIM}} = (4.328125 \pm 10^{-3}) \pm 10^{-3}i.$$

Example 6: Let D be given by

$$(0,1) \times (0,1) \times (0,1) \setminus [0,1/2] \times [0,1/2] \times [0,1/2]$$

with the index of refraction $n = 16$. The generalized eigenvalue problem is 13335×13335. Let $S = [4,5] \times [-1/2, 1/2]$. The eigenvalues in S are

$$\lambda_1 = 12.249750, \quad \lambda_2 = 13.102771.$$

RIM outputs

$$\lambda_1^{\mathbf{RIM}} = (12.249725 \pm 10^{-3}) \pm 10^{-3}i$$

and

$$\lambda_2^{\mathbf{RIM}} = (13.102753 \pm 10^{-3}) \pm 10^{-3}i.$$

For above examples in 2D or 3D, **RIM** returns all eigenvalues in a given region correctly.

2.4.2 Robustness

We demonstrate the robustness of **RIM** related to the use of one random vector and one quadrature point on each side of the rectangle S. We test three cases.

(**i**) S contains eigenvalues. Let

$$S_1 = [3.9, 4.1] \times [-0.1, 0.1], \quad S_2 = [24.1, 24.2] \times [5.6, 5.7]$$

for **Example 1**, and

$$S_3 = [6.04, 6.06] \times [-0.01, 0.01], \quad S_4 = [20.5, 20.6] \times [5.1, 5.2]$$

for **Example 2**. Each region has an eigenvalue inside. We compute the indicators for 100 random vectors. The results are shown in Table 2.1. The first column shows the regions. The second, third, fourth, and fifth columns are the average, minimum, maximum, and the standard deviation of the indicators, respectively. We can see that different random vectors give similar indicators. The standard deviation is very small. In other words, the algorithm is tested 100 times using different random vectors. **RIM** produces the correct results.

Table 2.1: The indicators for different regions with eigenvalues inside.

S	average	min	max	std.
S_1	0.63662546	0.63662432	0.63662669	2.42494379e-07
S_2	0.82076270	0.82076270	0.82076270	3.48933530e-11
S_3	0.63667811	0.63662296	0.63674302	4.23597573e-05
S_4	0.53606809	0.53606809	0.53606809	5.68226051e-11

Remark 2.4.0.2. Table 2.1 shows that there is no big difference between choosing one random vector and many different random vectors. Note that **RIM** is not a subspace method. There is no need to know how many eigenvalues inside Γ and choose more random vectors to generate a subspace problem.

(**ii**) S contains no eigenvalue. Let $S_5 = [3.7, 3.9] \times [-0.1, 0.1]$ and $S_6 = [24.0, 24.1] \times [5.6, 5.7]$ for **Example 1**. Let $S_7 = [6.02, 6.04] \times [-0.01, 0.01]$ and $S_8 = [20.4, 20.5] \times [5.1, 5.2]$ for **Example 2**. These regions do not have eigenvalues inside. Again, we test the algorithm 100 times. Each time, we use one random vector, which is different from time to time. In Table 2.2, it can be seen that the indicators are very small, indicating that there are no eigenvalue(s) in these regions.

(**iii**) S has an eigenvalue on its edge or at a corner. For **Example 1**, we choose two rectangles

$$S_{13} = [3.99, 4.00] \times [-0.01, 0.00] \quad \text{and} \quad S_{14} = [3.99, 4.00] \times [0.00, 0.01]$$

Table 2.2: The indicators for different regions with no eigenvalues inside.

S	average	min	max	std.
S_5	0.04778437	0.04778398	0.04778539	1.48221826e-07
S_6	0.02227906	0.02227906	0.02227906	6.92810353e-12
S_7	0.04143107	0.03354195	0.04701297	4.44534110e-03
S_8	0.01615294	0.01615294	0.01615294	4.94631163e-11

sharing an edge. Since the eigenvalue $\lambda_1 = 3.994690$ is real, it is on the boundary of S. In Table 2.3, we show the indicators. We can see that both regions are admissible. Next, we choose S_{15} and S_{16} such that the sharing edge goes through an complex eigenvalue. The indicator for S_{16} is smaller than 1/10. However, this is fine since S_{15} is admissible and we will catch the eigenvalue.

Table 2.3: The indicators when the eigenvalue is on the edge of the search region.

domain	indicator
$S_{13} = [3.99, 4.00] \times [-0.01, 0.00]$	0.52275012
$S_{14} = [3.99, 4.00] \times [0.00, 0.01]$	0.52275012
$S_{15} = [24.158813, 24.17] \times [5.68, 5.70]$	0.48810370
$S_{16} = [24.15, 24.158813] \times [5.68, 5.70]$	0.08569820

Next we consider the case when an eigenvalue is a corner of the search region. We know from above that search regions S_{17}, S_{18}, S_{19}, and S_{20} (see Table 2.4) sharing a corner, which is an eigenvalue.

The choice of the threshold value δ_0 is important to the robustness of **RIM**. Note that approximation of the contour integral, including the quadrature and the linear solver, introduces some errors, especially when eigenvalues are close to Γ or even on

Γ. In fact, this is the case whenever the search region is close to the eigenvalues. The algorithm uses the threshold value 1/10 based on experiments. The above examples show that the choice is effective.

Table 2.4: The indicators when the eigenvalue is a corner of the search region.

domain	indicator
$S_{17} = [3.994539, 4.01] \times [-0.01, 0.0]$	0.70164096
$S_{18} = [3.98, 3.994539] \times [-0.01, 0.0]$	0.91502267
$S_{19} = [3.98, 3.994539] \times [0.00, 0.01]$	0.25047335
$S_{20} = [3.994539, 4.01] \times [0.00, 0.01]$	0.25047335
$S_{21} = [24.152, 24.158567] \times [5.688, 5.690114]$	0.43892705
$S_{22} = [24.152, 24.158567] \times [5.690114, 5.700]$	0.12732395
$S_{23} = [24.158567, 24.161] \times [5.690114, 5.700]$	0.12732395
$S_{24} = [24.158567, 24.161] \times [5.688, 5.690114]$	0.19531957

2.4.3 Self-correction Property

The choice of threshold value is related to a nice property of **RIM**, which we call the *self-correction property*. Consider the case when S is not admissible but close to an eigenvalue. At some quadrature points, the linear systems are ill-conditioned. In addition, the quadrature rule might not be sufficiently accurate. **RIM** might take such region as admissible at first. Fortunately, after a few subdivisions, **RIM** discards these regions. We demonstrate this interesting *self-correction property* using two example.

We use matrices A and B from **Example 1** and focus on the eigenvalue 3.994539. We choose the initial search region $S = [4.0, 4.2] \times [0, 0.2]$. Note that there is no eigenvalue in S and 3.994539 is right outside S. At first, **RIM** computes

$$\delta_S = 0.11666587, \tag{2.4.11}$$

indicating that S is admissible. **RIM** continues to explore S by partitioning it into four rectangles

$$S_1^1 = [4.0, 4.1] \times [0, 0.1], \quad S_2^1 = [4.0, 4.1] \times [0.1, 0.2],$$
$$S_3^1 = [4.1, 4.2] \times [0, 0.2], \quad S_4^1 = [4.1, 4.2] \times [0.1, 0.2].$$

The indicators are

$$\delta_{S_1^1} = 0.10687367, \quad \delta_{S_2^1} = 0.00609138,$$
$$\delta_{S_3^1} = 0.00561028, \quad \delta_{S_4^1} = 0.00182170.$$

RIM discards S_2^1, S_3^1, and S_4^1 and retains S_1^1 as admissible.

The four rectangles by partitioning S_1^1 are

$$S_1^2 = [4.0, 4.05] \times [0.0, 0.05], \quad S_2^2 = [4.0, 4.05] \times [0.05, 0.10],$$
$$S_3^2 = [4.05, 4.10] \times [0.0, 0.05], \quad S_4^2 = [4.05, 4.10] \times [0.05, 0.10].$$

The indicators are

$$\delta_{S_1^2} = 0.08957100, \quad \delta_{S_2^2} = 0.00579253,$$
$$\delta_{S_3^2} = 0.00494816, \quad \delta_{S_4^2} = 0.00169435.$$

At this stage, **RIM** discards all the regions. Let us see one more level. Suppose S_1^2 is subdivided into

$$S_1^3 = [4.0, 4.025] \times [0, 0.025], \quad S_2^3 = [4.0, 4.025] \times [0.025, 0.05],$$
$$S_3^3 = [4.025, 4.05] \times [0, 0.025], \quad S_4^3 = [4.025, 4.05] \times [0.025, 0.05],$$

with the following indicators

$$\delta_{S_1^3} = 0.06258907, \quad \delta_{S_2^3} = 0.00519080,$$

$$\delta_{S_3^3} = 0.00388825, \quad \delta_{S_4^3} = 0.00146650.$$

RIM will eventually discard all the subregions and concludes that there are no eigenvalues in S.

The same experiment is conducted for $S = [24.16, 24.96] \times [5.30, 6.10]$, a search region close to a complex eigenvalue $\lambda = 24.158567 + 5.690308i$. Indicators are given in Table. 2.5. **RIM** does eventually conclude that there are no eigenvalues in the region.

Table 2.5: The indicators for $S = [24.16, 24.96] \times [5.30, 6.10]$.

$S_1^1 = [24.16, 24.56] \times [5.30, 5.70]$	0.825
$S_2^1 = [24.16, 24.56] \times [5.70, 6.10]$	0.195
$S_3^1 = [24.56, 24.96] \times [5.30, 5.70]$	5.418e-11
$S_4^1 = [24.56, 24.96] \times [5.70, 6.10]$	4.119e-11
$S_1^2 = [24.16, 24.36] \times [5.30, 5.50]$	9.216e-11
$S_2^2 = [24.16, 24.36] \times [5.50, 5.70]$	0.368
$S_3^2 = [24.36, 24.56] \times [5.30, 5.50]$	8.712e-14
$S_4^2 = [24.36, 24.56] \times [5.50, 5.70]$	5.870e-11
$S_1^3 = [24.16, 24.26] \times [5.50, 5.60]$	1.742e-11
$S_2^3 = [24.16, 24.26] \times [5.60, 5.70]$	0.781
$S_3^3 = [24.26, 24.36] \times [5.50, 5.60]$	1.476e-13
$S_4^3 = [24.26, 24.36] \times [5.60, 5.70]$	6.755e-11
$S_1^4 = [24.16, 24.21] \times [5.60, 5.65]$	6.558e-10
$S_2^4 = [24.16, 24.21] \times [5.65, 5.70]$	0.280
$S_3^4 = [24.21, 24.26] \times [5.60, 5.65]$	1.378e-13
$S_4^4 = [24.21, 24.26] \times [5.65, 5.70]$	8.229e-11
$S_1^5 = [24.16, 24.185] \times [5.65, 5.675]$	1.159e-08
$S_2^5 = [24.16, 24.185] \times [5.675, 5.70]$	0.156
$S_3^5 = [24.185, 24.21] \times [5.65, 5.675]$	4.000e-13
$S_4^5 = [24.185, 24.21] \times [5.675, 5.70]$	8.648e-11
$S_1^6 = [24.16, 24.185] \times [5.65, 5.675]$	5.574e-06
$S_2^6 = [24.16, 24.1725] \times [5.6875, 5.70]$	0.095
$S_3^6 = [24.185, 24.21] \times [5.65, 5.675]$	4.304e-12
$S_4^6 = [24.185, 24.21] \times [5.675, 5.70]$	2.628e-11

2.4.4 Close Eigenvalues

RIM can separates nearby eigenvalues provided the precision ϵ is less than the distance between them. For **Example 1**, there are two close eigenvalues

$$\lambda_1 = 6.935054, \quad \lambda_2 = 6.939719.$$

With $\epsilon = 3.0 \times 10^{-2}$, **RIM** fails to separate the eigenvalues and outputs only one eigenvalue

$$\lambda_1^{\mathbf{RIM}} = 6.942500 \pm 3 \times 10^{-2}(1 \pm i).$$

However, with $\epsilon = 10^{-4}$, **RIM** separates the eigenvalues

$$\lambda_1^{\mathbf{RIM}} = 6.935127 \pm 10^{-4}(1 \pm i),$$
$$\lambda_2^{\mathbf{RIM}} = 6.939717 \pm 10^{-4}(1 \pm i).$$

2.5 Conclusion

In this chapter, we propose a novel recursive integral method **RIM** for eigenvalue problems and employ it to compute transmission eigenvalues. The method can effectively find all eigenvalues in a region with no a priori spectrum information. The key difference between **RIM** and other contour integral based methods in the literature is that **RIM** only tests if a region contains eigenvalues or not.

In the next few chapters, we are going to discuss some improvements on **RIM**.

Chapter 3

Recursive Integral Method with Cayley Transformation [1]

Abstract

In the last chapter, we proposed the recursive integral method (**RIM**) for computing all eigenvalues in a region on the complex plane. In this chapter, we propose an improved version of **RIM** for non-Hermitian eigenvalue problems. Using Cayley transformation and Arnoldi's method, the computation cost is reduced significantly. Effectiveness and efficiency of the new method are demonstrated by numerical examples and compared with 'eigs' in Matlab.

3.1 Introduction

We consider the non-Hermitian eigenvalue problem

$$Ax = \lambda Bx, \tag{3.1.1}$$

where A and B are $n \times n$ large sparse matrices. Here B can be singular. Such eigenvalue problems arise in many scientific and engineering applications [33, 3, 1] as

[1]This chapter has been published as an article in Numerical Linear Algebra with Applications. https://doi.org/10.1002/nla.2199

well as in emerging areas such as data analysis in social networks [34].

The problem of interest in this chapter is to find (all) eigenvalues with less computation resource in a given region S on the complex plane $\mathbb{C}$ without any spectral information, i.e., the number and distribution of eigenvalues in S are not known.

In last chapter, we developed an eigenvalue solver **RIM** (recursive integral method). **RIM**, which is essentially different from all the existing eigensolvers, is based on spectral projection and domain decomposition. As introduced in chapter 2, the indicator is defined as $\delta_S = \left| P\left(\frac{P\boldsymbol{f}}{|P\boldsymbol{f}|}\right) \right|$ in (2.3.10). To compute δ_S, one needs to solve many linear systems

$$(A - z_j B)\boldsymbol{x}_j = \boldsymbol{f} \tag{3.1.2}$$

parameterized by z_j. In the original **RIM** , the Matlab linear solver '\' is used to solve (3.1.2). This is certainly not efficient.

Thus in this chapter, we propose a new version of **RIM**, called **RIM-C**, to improve the efficiency. The contributions include: 1) Cayley transformation and Arnoldi's method to speedup linear solves for the parameterized system (3.1.2); and 2) a new indicator to improve the robustness and efficiency. The rest of the chapter is arranged as follows. In Section 3.2, we present how to incorporate Cayley transformation and the Arnoldi's method into **RIM**. In Section 3.3, we introduce a new indicator to decide if a region contains eigenvalues. Section 3.4 contains the new algorithm and some implementation details. Numerical examples are presented in Section 3.5. We end up the chapter with some conclusions and future works in Section 3.6.

3.2 Cayley Transformation and Arnoldi's Method

3.2.1 Cayley Transformation

The computation cost of **RIM** mainly comes from solving the linear systems (3.1.2) to compute the spectral projection $P\boldsymbol{f}$. In particular, when the method zooms in around an eigenvalue, it needs to solve linear systems for many close z_j's. This is done one by one in the first version of **RIM** [35]. It is clear that the computation cost will be greatly reduced if one can take the advantage of the parametrized linear systems of same structure.

Without loss of generality, we consider a family of linear systems

$$(A - zB)\boldsymbol{x} = \boldsymbol{f}, \tag{3.2.3}$$

where z is a complex number. When B is nonsingular, multiplication of B^{-1} on both sides of (3.2.3) leads to

$$(B^{-1}A - zI)\boldsymbol{x} = B^{-1}\boldsymbol{f}. \tag{3.2.4}$$

Given a matrix M, a vector $\boldsymbol{b}$, and a non-negative integer m, the Krylov subspace is defined as

$$K_m(M; \boldsymbol{b}) := \text{span}\{\boldsymbol{b}, M\boldsymbol{b}, \ldots, M^{m-1}\boldsymbol{b}\}. \tag{3.2.5}$$

The shift-invariant property of Krylov subspaces says that

$$K_m(aM + bI; \boldsymbol{b}) = K_m(M; \boldsymbol{b}), \tag{3.2.6}$$

where a and b are two scalars. Thus the Krylov subspace of $B^{-1}A - zI$ is the same as $B^{-1}A$, which is independent of z.

The above derivation fails when B is singular. Fortunately, this can be fixed by Cayley transformation [36]. Assume that σ is not a generalized eigenvalue and $\sigma \neq z$. Multiplying both sides of (3.2.3) with

$$(A - \sigma B)^{-1}, \tag{3.2.7}$$

one obtains that

$$\begin{aligned} (A - \sigma B)^{-1}\boldsymbol{f} &= (A - \sigma B)^{-1}(A - zB)\boldsymbol{x} \\ &= (A - \sigma B)^{-1}(A - \sigma B + (\sigma - z)B)\boldsymbol{x} \\ &= (I + (\sigma - z)(A - \sigma B)^{-1}B)\boldsymbol{x}. \end{aligned}$$

Let $M = (A - \sigma B)^{-1}B$ and $\boldsymbol{b} = (A - \sigma B)^{-1}\boldsymbol{f}$. Then (3.2.3) becomes

$$(I + (\sigma - z)M)\boldsymbol{x} = \boldsymbol{b}. \tag{3.2.8}$$

From (3.2.6), the Krylov subspace $(I + (\sigma - z)M)$ is the same as $K_m(M; \boldsymbol{b})$.

3.2.2 Analysis of the Pre-conditioners

Now we look at the connection between two pre-conditioners B^{-1} and $(A - \sigma B)^{-1}$. Assume that B is non-singular. Let λ be an eigenvalue of $B^{-1}A$. Then $\theta = \dfrac{\lambda - z}{\lambda - \sigma}$ is an eigenvalue of

$$(A - \sigma B)^{-1}(A - zB).$$

The spectrum of $B^{-1}A$ might spread over the complex plane such that Krylov subspace based iterative methods may not converge. However, after Cayley transformation, when λ becomes large, θ will cluster around 1 (see Fig. 3.1 for matrices A and B of **Example 1** in Section 5). Similar result holds when B is singular. Note that when λ approaches σ, θ will be very large in magnitude. When λ approaches z, θ goes to zero. When λ is away from σ and z, θ is $O(1)$. The key here is that the spectrum of (3.2.8) has a cluster of eigenvalues around 1 and only a few isolated eigenvalues, which favors fast convergence in Krylov subspace.

3.2.3 Arnoldi Method for Linear Systems

The computation cost can be significantly reduced by exploiting (3.2.8). Consider the orthogonal projection method for

$$M\boldsymbol{x} = \boldsymbol{b}.$$

Let the initial guess be $\boldsymbol{x}_0 = \mathbf{0}$. One seeks an approximate solution $\boldsymbol{x}_m$ in $K_m(M; \boldsymbol{b})$ of dimension m by imposing the Galerkin condition [37]

$$(\boldsymbol{b} - M\boldsymbol{x}_m) \perp K_m(M; \boldsymbol{b}). \tag{3.2.9}$$

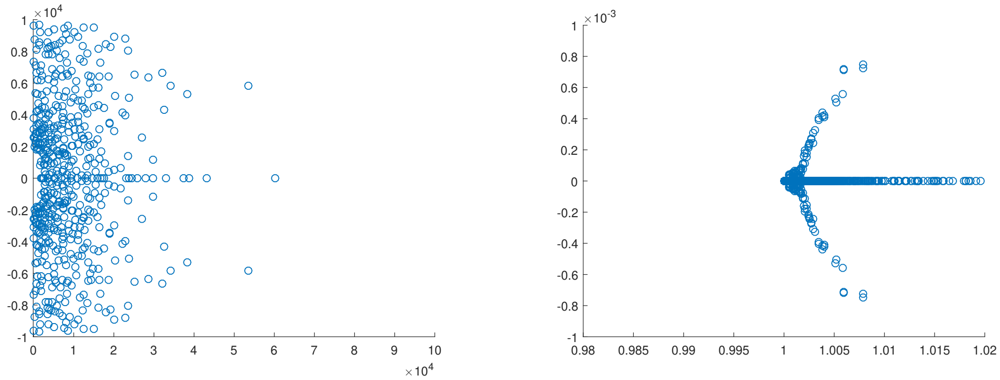


Figure 3.1: Matrices A and B are from Example 1 in Section 5. Left: Spectrum of original problem. Right: Spectrum after Cayley transformation.

The basic Arnoldi's process (Algorithm 6.1 of [3]) is as follows.

1. Choose a vector $\boldsymbol{v}_1$ of norm 1

2. for $j = 1, 2, \ldots, m$

   - $h_{ij} = (M\boldsymbol{v}_j, \boldsymbol{v}_i), \quad i = 1, 2, \ldots, j,$

   - $\boldsymbol{w}_j = M\boldsymbol{v}_j - \sum_{i=1}^{j} h_{ij}\boldsymbol{v}_i,$

   - $h_{j+1,j} = \|\boldsymbol{v}_j\|_2$, if $h_{j+1,j} = 0$ stop

   - $\boldsymbol{v}_{j+1} = \boldsymbol{w}_j / h_{j+1,j}.$

Let V_m be the $n \times m$ orthogonal matrix with column vectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m$ and H_m be the $m \times m$ Hessenberg matrix whose nonzero entries $h_{i,j}$ are defined as above. From Proposition 6.6 of [3], one has that

$$MV_m = V_m H_m + \boldsymbol{v}_{m+1} h_{m+1,m} \boldsymbol{e}_m^T \tag{3.2.10}$$

such that

$$\text{span}\{\text{col}(V_m)\} = K_m(M; \boldsymbol{b}).$$

Let $\boldsymbol{x}_m = V_m \boldsymbol{y}$. The Galerkin condition (3.2.9) becomes

$$V_m^T \boldsymbol{b} - V_m^T M V_m \boldsymbol{y} = \boldsymbol{0}. \tag{3.2.11}$$

Since $V_m^T M V_m = H_m$ (see Proposition 6.5 of [37]), the following holds:

$$H_m \boldsymbol{y} = V_m^T \boldsymbol{b}.$$

From the construction of V_m, $\boldsymbol{v}_1 = \dfrac{\boldsymbol{b}}{\|\boldsymbol{b}\|_2}$. Let $\beta = \|\boldsymbol{b}\|_2$. Then

$$\boldsymbol{y} = \beta H_m^{-1} \boldsymbol{e}_1.. \tag{3.2.12}$$

Consequently, the residual of the approximated solution $\boldsymbol{x}_m$ can be written as

$$\|\boldsymbol{b} - M\boldsymbol{x}_m\|_2 = h_{m+1,m} |\boldsymbol{e}_m^T \boldsymbol{y}|. \tag{3.2.13}$$

Due to the shift invariant property, one has that

$$\{I + (\sigma - z)M\} V_m = V_m (I + (\sigma - z) H_m) + (\sigma - z) \boldsymbol{v}_{m+1} h_{m+1,m} \boldsymbol{e}_m^T. \tag{3.2.14}$$

By imposing a Galerkin condition similar to (3.2.9), we have that

$$V_m^T \boldsymbol{b} - V_m^T \{I + (\sigma - z)M\} V_m \boldsymbol{y} = 0, \tag{3.2.15}$$

which implies

$$\{I + (\sigma - z) H_m\} \boldsymbol{y} = \beta \boldsymbol{e}_1. \tag{3.2.16}$$

From (3.2.13), one has that

$$\|\boldsymbol{b} - \{I + (\sigma - z)M\} \boldsymbol{x}_m\|_2 = (\sigma - z) h_{m+1,m} |\boldsymbol{e}_m^T \boldsymbol{y}|. \tag{3.2.17}$$

Matrix M is an $n \times n$ matrix and H_m is an $m \times m$ upper Hessenberg matrix such that $m \ll n$. Once H_m and V_m are constructed by Arnoldi's process, they can be used to solve (3.2.16) for different z's with residual given by (3.2.17). The residual can be monitored with a little extra cost.

Next we explain how the Arnoldi's process is incorporated in **RIM**. To solve (3.1.2) for quadrature points z_j's, one chooses a proper shift σ. Following (3.2.8), one has that

$$(I + (\sigma - z_j)M)\boldsymbol{x}_j = \boldsymbol{b}, \tag{3.2.18}$$

where $M = (A - \sigma B)^{-1}B$ and $\boldsymbol{b} = (A - \sigma B)^{-1}\boldsymbol{f}$.

From (3.2.14) and (3.2.16),

$$\begin{aligned} \boldsymbol{y}_j &= \beta(I + (\sigma - z_j)H_m)^{-1}\boldsymbol{e}_1, && (3.2.19) \\ \boldsymbol{x}_j &\approx V_m \boldsymbol{y}_j, \\ P\boldsymbol{f} &\approx \frac{1}{2\pi i}\sum w_j V_m \boldsymbol{y}_j. && (3.2.20) \end{aligned}$$

Hence the Krylov subspace for $M = (A - \sigma B)^{-1}B$ can be used to solve many linear systems associated with z_j's close to σ.

3.3 An Efficient Indicator

Another critical problem of **RIM** is to how to define the indicator δ_S. As seen above, the indicator in [35] defined by (2.3.10) is to project a random vector twice. One needs to solve linear systems with different right hand sides, i.e., $\boldsymbol{f}$ and $P\boldsymbol{f}/|P\boldsymbol{f}|$. Consequently, two Krylov subspaces, rather than one, are constructed for a single shift σ.

In this section, we propose a new indicator that avoids the construction of two Krylov subspaces. The indicator stills needs to resolve the two problems (P1 and P2) in Section 1. The idea is to approximate $|P\boldsymbol{f}|$ with different sets of trapezoidal quadrature points by taking the advantage of the Cayley transformation and Arnoldi's method discussed in the previous section.

Let $P\boldsymbol{f}|_n$ be the approximation of $P\boldsymbol{f}$ with n quadrature points. It is well-known that trapezoidal quadratures of a periodic function converges exponentially [38] i.e.,

$$|P\boldsymbol{f} - P\boldsymbol{f}|_n| = O(e^{-Cn}),$$

where C is a constant depending on $\boldsymbol{f}$. The spectral projection satisfies

$$P\boldsymbol{f}|_n \begin{cases} \neq \mathbf{0} & \text{if there are eigenvalues inside } S, \\ \approx \mathbf{0} & \text{no eigenvalue inside } S. \end{cases}$$

For a large enough n_0, one has that

$$\frac{|P\boldsymbol{f}|_{2n_0}|}{|P\boldsymbol{f}|_{n_0}|} = \begin{cases} \dfrac{|P\boldsymbol{f}| + O(e^{-C2n})}{|P\boldsymbol{f}| + O(e^{-Cn})} & \text{if there are eigenvalues inside } S, \\ \dfrac{O(e^{-C2n})}{O(e^{-Cn})} = O(e^{-Cn}) & \text{no eigenvalue inside } S. \end{cases}$$

The new indicator is set to be

$$\delta_S = |P\boldsymbol{f}_{2n_0}|/|P\boldsymbol{f}_{n_0}|. \tag{3.3.21}$$

A threshold value δ_0 is also needed to decide if there exists eigenvalue in S or not. If $\delta_S > \delta_0 := 0.2$, S is said to be admissible, i.e., there exists eigenvalue(s) in S. The value 0.2 is chosen based on numerical experimentation. Due to (3.2.19) - (3.2.20), the computation cost to evaluate the new indicator is not expensive.

3.4 The New Algorithm

Now we are ready to give the algorithm in detail. It starts with several shifts σ's distributed in S uniformly. The associated Krylov subspaces $K_m(M; \boldsymbol{b})$ are constructed and stored. For a quadrature point z, the algorithm first attempts to solve the linear system (3.2.3) using the Krylov subspace with shift σ closest to z. If the residual is larger than the given precision ϵ, a Krylov subspace with a new shift σ is constructed, stored and used to solve the linear system. Briefly speaking, the algorithm constructed some Krylov subspaces with different σ's. These subspaces are then used to solve the linear system for all quadrature points z_j's. From (3.2.19) and (3.2.20), instead of solving a family of linear systems of size n, the algorithm solves linear systems of reduced size m for most z_j's. This is the key idea to speed up **RIM**. We denote this improved version of **RIM** by **RIM-C** (**RIM** with Cayley transformation).

Given a search region S and a normalized random vector $\boldsymbol{f}$, we compute the indicator δ_S using (3.3.21). Without loss of generality, S is assumed to be a square. We set $n_0 = 4$ in (3.3.21). If $\delta_S > 0.2$, S is divided uniformly into 4 regions. The indicators of these regions are computed. This process continues until the size of the region is smaller than d_0.

Algorithm RIM-C:

RIM-C$(A, B, S, \boldsymbol{f}, d_0, \epsilon, \delta_0, m, n_0)$

Input:

- A, B: $n \times n$ matrices
- S: search region in $\mathbb{C}$
- $\boldsymbol{f}$: a random vector
- d_0: precision
- ϵ: residual threshold
- δ_0: indicator threshold
- m: size of Krylov subspace
- n_0: number of quadrature points

Output:

- generalized eigenvalues inside S

1. Choose several σ's uniformly in S and construct Krylov subspaces
2. Compute δ_S using (3.3.21).

   Let z be a quadrature point.

   - Check if the linear system can be solved using the existing Krylov subspaces with residual less than ϵ.

- Otherwise, choose a new σ, construct a new Krylov subspace to solve the linear system.

3. Decide if each S contains eigenvalues(s).

- If $\delta_S = \dfrac{|P\boldsymbol{f}|_{2n_0}|}{|P\boldsymbol{f}|_{n_0}|} < \delta_0$, exit.
- Compute the size of S, $h(S)$.

- If $h(S) > \epsilon_0$, uniformly partition S_i into subregions $S_j, j = 1, \dots 4$

  for $j = 1$ to 4

  call **RIM-C**$(A, B, S_j, \boldsymbol{f}, d_0, \epsilon, \delta_0, m, n_0)$

  end

- Otherwise, output the eigenvalue λ and exit.

3.5 Numerical Examples

In this section, **RIM-C** (implemented in Matlab) is employed to compute all the eigenvalues in a given region. To the authors' knowledge, there exists no eigensolver doing exactly the same thing. We compare **RIM-C** with 'eigs' in Matlab (IRAM: Implicitly Restarted Arnoldi Method [39]). Although the comparison seems to be unfair to both methods, it gives some idea about the performance of **RIM-C**.

The matrices for **Examples 1-5** come from a finite element discretization of the transmission eigenvalue problem [2, 9] using different mesh size h. Therefore, the spectra of these problems are similar. For Matlab function 'eigs(A,B,K,SIGMA)', 'K' and 'SIGMA' denote the number of eigenvalues to compute and the *shift*, respectively. For **RIM-C**, the size of Krylov space is set to be $m = 50$, $d_0 = 10^{-9}$, $\epsilon = 10^{-10}$, $\delta_0 = 0.2$, and $n_0 = 4$. All the examples are computed on a Macbook pro with 16 Gb memory and 3 GHz Intel Core i7.

Example 1: The matrices A and B are 1018×1018 (mesh size $h \approx 0.1$). The search region $S = [1, 11] \times [-1, 1]$. For 'eigs', the 'shift' is set to be 5.5. For this problem, it is known that there exist 5 eigenvalues in S. Therefore, 'K' is set to be 5. Note that **RIM-C** does not need this information. The results are shown in Table 3.1. Both **RIM-C** and 'eigs' compute 5 eigenvalues and they are consistent. 'eigs' uses less time than **RIM-C**.

Table 3.1: Eigenvalues computed and CPU time by **RIM-C** and 'eigs' for **Example 1**.

	RIM-C	'eigs'
Eigenvalues	**3.9945390188**48445	**3.9945390188**56096
	6.939719143800903	**6.9397191438**04773
	6.935053985873570	**6.9350539858**44678
	10.654665853490588	**10.6546658534**41946
	10.658706024650019	**10.6587060246**09756
CPU time	0.284922s	0.247310s

Example 2: Matrices A and B are 4066×4066 (mesh size $h \approx 0.05$). Let $S = [20, 30] \times [-6, 6]$. For 'eigs', '*shift*' is set to be 25. Again, it is known in advance that there are 3 eigenvalues in S. Hence 'K' is set to be 3. The results are shown in Table 3.2. Both methods compute same eigenvalues and 'eigs' is faster.

Example 3: Matrices A and B are 16258×16258 matrices (mesh size $h \approx 0.025$). Let $S = [0, 20] \times [-6, 6]$. There are 10 eigenvalues in S. It is well-known that the performance of 'eigs' is highly dependent on 'shift'. In Table 3.3, we show the time used by **RIM-C** and 'eigs' with different shifts '*shift* =5, 10, 15'. Notice that when the shift is not *good*, 'eigs' uses much more time. In practice, *good* shifts are not known in advance.

Table 3.2: Eigenvalues computed and CPU time by **RIM-C** and 'eigs" for **Example 2**.

	RIM-C	'eigs'
Eigenvalues	**23.803023938**395199	**23.803023938**403236
	± **5.682304314876**092i	± **5.6823043148**40053i
	24.737027497006540	**24.737027497003**453
	24.750959635036583	**24.7509596350**22376
	25.278145187465789	**25.2781451874**57707
	25.284501515028143	**25.2845015150**36474
CPU time	0.558687s	0.333513s

Table 3.3: CPU time used by **RIM** and 'eigs' with different shifts for **Example 3**.

	RIM-C	'eigs' shift=5	'eigs' shift=10	'eigs' shift=15
CPU time	2.571800s	0.590186	**7.183679**s	0.392902s

Example 4: We consider a larger problem: A and B are 260098×260098. Let $S = [0, 20] \times [-6, 6]$ (mesh size $h \approx 0.00625$). There are 17 eigenvalues in S. The results are in Table 3.4. This example, again, shows that for larger problems without any spectrum information, the performance of **RIM-C** is quite stable and consistent. However, the performance of 'eigs' varies a lot with different '*shifts*'.

Table 3.4: CPU time used by **RIM** and 'eigs' with different *shifts* for **Example 4**.

	RIM-C	'eigs' shift=5	'eigs' shift=10
CPU time	104.228413s	**1696.703477**s	**272.506573**s

Example 5: This example demonstrates the effectiveness and robustness of the new indicator. The same matrices in **Example 3** (16258×16258) are

used. Consider three regions S_1, S_2 and S_3. $S_1 = [18.4, 18.8] \times [-0.2, 0.2]$ has one eigenvalue inside. $S_2 = [14.6, 14.8] \times [-0.1, 0.1]$ has two eigenvalues inside. $S_3 = [19.7, 19.9] \times [-0.1, 0.1]$ contains no eigenvalue. Table 3.5 shows the indicators of these three regions computed using (3.3.21). It is seen that the indicator is different when there are eigenvalues inside the region and when there are no eigenvalues.

Table 3.5: Indictors: S_1 and S_2 contain at least one eigenvalue, S_3 contains no eigenvalue.

# of quadrature points	$P\boldsymbol{f}_{S_1}$	$P\boldsymbol{f}_{S_2}$	$P\boldsymbol{f}_{S_3}$
4	0.0210361614	0.0002565318	0.0011737026
8	0.0209817055	0.0002585042	0.0000442384
δ_S	0.997411	0.992370	0.037691

Table 3.6 shows the means, minima, maxima, and standard deviations of indicators of these three regions computed using 100 random vectors. The indicators are consistent for different random vectors.

Table 3.6: Means, minima, maxima, and standard deviations of indicators using 100 random vectors.

S	mean	min.	max.	std. dev.
S_1	0.99848393687	0.66250246918	1.43123449889	0.08740952445
S_2	0.99926772105	0.92600650392	1.14648387384	0.01788832121
S_3	0.03763601782	0.03734608324	0.03775912970	0.00010228556

Example 6: The last example shows the potential of **RIM-C** to treat large matrices. The sparse matrices are of $\mathbf{15,728,640} \times \mathbf{15,728,640}$ arising from a finite element discretization of localized quantum states in

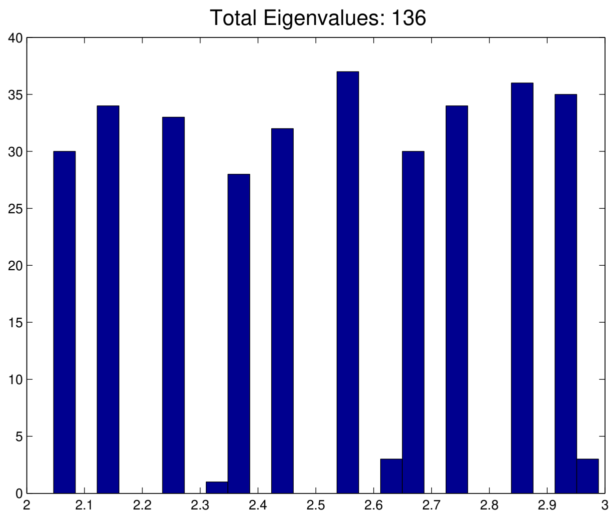


Figure 3.2: Distribution of eigenvalues in $(2, 3)$ for **Example 6**.

random media [40]. **RIM-C** computed 136 real eigenvalues in $(2, 3)$, shown in the right picture of Fig. 3.2.

3.6 Conclusions

This purposes of this chapter is to compute (all) the eigenvalues of a large sparse non-Hermitian problem in a given region. We propose a new eigensolver **RIM-C**, which is an improved version of the recursive integral method using spectrum projection. **RIM-C** uses Cayley transformation and Arnoldi method to reduce the computation cost.

To the authors' knowledge, **RIM-C** is the only eigensolver for this particular purpose. As we mentioned, the comparison of **RIM-C** and 'eigs' is unfair to both methods. However, the numerical results do show that **RIM-C** is effective and has the potential to treat large scale problems.

In next chapter, we are going to introduce the multilevel spectral indicator method (**SIM-M**) based on **RIM-C** for its efficient memory.

Chapter 4

A Memory Efficient Multilevel Spectral Indicator Method [1]

Abstract

In last chapter, we proposed the improved version of **RIM** ,the **RIM-M** for computing all eigenvalues in a region on the complex plane. In this chapter, by a special way of using Cayley transformation and Krylov subspaces, a memory efficient multilevel eigensolver for large sparse eigenvalue problems is proposed. This method is fast, uses little memory, and is particularly suitable to compute many eigenvalues. The method is implemented in Matlab and tested by various matrices.

Keywords: Spectral indicator method, Non-Hermitian sparse eigenvalue problems

4.1 Introduction

Many efficient eigensolvers are proposed in literature for large sparse Hermitian (or symmetric) matrices. In contrast, for non-Hermitian problems,

[1]This chapter has been submitted.

there exist fewer methods including the Arnoldi method, Lanczos method and Jacobi-Davidson method [41]. Unfortunately, these methods are still far from satisfactory as pointed out in [3]: "*In essence what differentiates the Hermitian from the non-Hermitian eigenvalue problem is that in the first case we can always manage to compute an approximation whereas there are non-symmetric problems that can be arbitrarily difficult to solve and can essentially make any algorithm fail.*"

Spectral projection is a classical tool in functional analysis to study, e.g., the spectrum of operators [28] and the finite element convergence theory for eigenvalue problems of partial differential equations [1]. It has been used to compute matrix eigenvalue problems in the method by Sakurai-Sugiura [29] and FEAST by Polizzi [4]. For example, FEAST uses spectral projection to build subspaces and thus can be viewed as a subspace method [42]. In contrast, SIMs only use the spectral projection to define indicators and do not actually solve any subspace problem.

In the last two chapters, we proposed RIM and its improved version RIM-M based the spectral projection. In this chapter, we propose a new member of SIMs, called SIM-M. Firstly, by proposing a new indicator, the memory requirement is significantly reduced and thus the computation of many eigenvalues of large matrices becomes realistic. Secondly, a new strategy to speedup the computation of the indicators is developed. Thirdly, other than the recursive calls in the first two members of SIMs [35, 43], a multilevel technique is used to further improve the efficiency. Moreover, a subroutine is added to find the multiplicities of the eigenvalues. The rest of the chapter is organized as follows. In Section 4.2, we propose a new eigensolver SIM-M with the above features. The algorithm and the implementation details are discussed as well. The proposed method is tested by various matrices in Section 4.3.

4.2 Multilevel Memory Efficient Method

In this section, we make several improvements of RIM-C and propose a multilevel memory efficient method, called SIM-M.

4.2.1 A New Memory Efficient Indicator

In view of (3.3.21), the computation of the indicator needs to store V_m. When R contains a lot of eigenvalues, many Krylov subspaces are needed and the method becomes memory intensive.

Definition 4.2.0.1. *A (square) region R is resolvable if the linear systems* (2.2.8) *associated with all the quadrature points for ∂R can be solved up to the given residual ϵ_0 using the Krylov subspace related to a shift σ. It is said to be unresolvable if R is not resolvable.*

Assume that R is resolvable. Since I_R in (3.3.21) is defined as a ratio, we propose a new indicator by dropping V_m in (3.2.20):

$$\tilde{I}_R = \frac{\|\sum_{j=1}^{2n_0} w_j \boldsymbol{y}_j\|}{\|\sum_{j=1}^{n_0} w_j \boldsymbol{y}_j\|}. \tag{4.2.1}$$

In fact, $I_R = \frac{\|V_m \sum_{j=1}^{2n_0} w_j \mathbf{y}_j\|}{\|V_m \sum_{j=1}^{n_0} w_j \mathbf{y}_j\|} = \frac{\|\sum_{j=1}^{2n_0} w_j \mathbf{y}_j\|}{\|\sum_{j=1}^{n_0} w_j \mathbf{y}_j\|} = \tilde{I}_R$ we have

$$\|V_m \sum_{j=1}^{n_0} w_j \mathbf{y}_j\| = (\sum_{j=1}^{n_0} w_j \mathbf{y}_j)^T V_m^T V_m \sum_{j=1}^{n_0} w_j \mathbf{y}_j = \|\sum_{j=1}^{n_0} w_j \mathbf{y}_j\|$$

since $V_m^T V_m$ is identity matrix from the construction of Krylov subspace. Consequently, there is no need to store V_m's ($n \times m$ matrices) but to store much smaller $m \times m$ ($m = O(1)$) matrices H_m's.

As before, we use a threshold to decide whether or not eigenvalues exist in R. From (3.2.20), if there are no eigenvalues in R, the indicator $I_R = O(e^{-Cn_0})$. In the experiments, we take $n_0 = 4$. Assume that $C = 1$, we would have that $I_R \approx 0.018$. It is reasonable to take $\delta_0 = 1/20$ as the threshold. However, it is still ad-hoc. Nonetheless, the numerical examples show that the choice is rather robust.

Definition 4.2.0.2. *A (square) region R is admissible if $I_R > \delta_0$.*

4.2.2 Speedup the Computation of Indicators

To check if a linear system (2.2.8) can be solved effectively using a Krylov space $K_m^\sigma(M; \boldsymbol{b})$, one need the compute the residual (3.2.17) for many z_j's. In the following, we propose a fast method for it. First rewrite (3.2.16) as

$$\left(\frac{1}{\sigma - z_j} I + H_m \right) \boldsymbol{y}_j = \frac{\beta}{\sigma - z_j} \boldsymbol{e}_1. \tag{4.2.2}$$

Assume that H_m has the following eigen-decomposition $H_m = PDP^{-1}$ where

$$D = \mathrm{diag}\{\lambda_1, \lambda_1, \ldots, \lambda_m\}.$$

Then (4.2.2) can be written as

$$P \left(\frac{1}{\sigma - z_j} I + D \right) P^{-1} \boldsymbol{y}_j = \frac{\beta}{\sigma - z_j} \boldsymbol{e}_1,$$

whose solution is simply

$$\begin{aligned} \boldsymbol{y}_j &= P \left(\frac{1}{\sigma - z_j} I + D \right)^{-1} P^{-1} \frac{1}{\sigma - z_j} \boldsymbol{e}_1 \\ &= P \left(I + (\sigma - z_j) D \right)^{-1} P^{-1} \boldsymbol{e}_1. \end{aligned}$$

Hence

$$\begin{aligned} \boldsymbol{e}_m^T \boldsymbol{y} &= \boldsymbol{e}_m^T P \left(I + (\sigma - z_j) D \right)^{-1} P^{-1} \boldsymbol{e}_1 \\ &= \boldsymbol{r}_m \Lambda \boldsymbol{c}_1, \end{aligned} \tag{4.2.3}$$

where $\boldsymbol{r}_m$ is the last row of P, $\boldsymbol{c}_1$ is first column of P^{-1}, and

$$\Lambda = \mathrm{diag} \left\{ \frac{1}{1 + (\sigma - z_j)\lambda_1}, \frac{1}{1 + (\sigma - z_j)\lambda_2}, \ldots, \frac{1}{1 + (\sigma - z_j)\lambda_m} \right\}.$$

In fact, this further reduces the memory requirement since only three $m \times 1$ vectors, $\boldsymbol{r}_m$, $\boldsymbol{c}_1$, and Λ are stored for each shift σ.

4.2.3 Multilevel Technique

Both RIM and RIM-C use recursive calls. However, a multilevel technique is more efficient and suitable for parallelization. In SIM-M, the following strategy is employed.

At level 1, R is divided uniformly into smaller squares $R_j^1, j = 1, \ldots, N^1$. Collect all quadrature points z_j^1's and solve the linear systems (2.2.8) accordingly. The indicators of R_j^1's are computed and squares containing eigenvalues are chosen. Indicators of the resolvable squares are computed. Squares containing eigenvalues are subdivided into smaller square. Squares that are not resolvable are also subdivided into smaller squares. These squares are left to the next level. At level 2, the same operation is carried out. The process stops at level K when. the size of the squares is smaller than the given precision.

4.2.4 Multiplicities of Eigenvalues

The first two members of SIMs only output the eigenvalues. A new function to find the multiplicities of the eigenvalues is integrated into SIM-M.

Definition 4.2.0.3. *An eigenvalue λ is said to be resolved by a shift σ if the small square at level K containing λ is resolvable using the Krylov subspace K_m^σ.*

When the eigenvalues are computed, a mapping from the set of eigenvalues Λ to the set of shifts Σ is also established. Hence, for a shift σ, one can find the set of all eigenvalues that are resolved by σ, denoted by

$$\Lambda_\sigma = \{\lambda_1, \ldots, \lambda_n\}.$$

For k random vectors $\boldsymbol{f}_1, \dots, \boldsymbol{f}_k$, generate k Krylov subspaces $K_m^\sigma(M, \boldsymbol{b}_i)$, $i = 1, \dots, k$. For each $\lambda \in \Lambda_\sigma$, compute the spectral projections of $\boldsymbol{f}_1, \dots, \boldsymbol{f}_k$ using the above Krylov subspaces. Then the number of significant singular values of the matrix $[P\boldsymbol{f}_1, \dots, P\boldsymbol{f}_k]$ is the multiplicity of λ.

Remark 4.2.0.1. In fact, the associated eigenvectors can be obtained with little extra cost by adding more quadrature points. However, it needs too much memory to store them.

4.2.5 Algorithm for SIM-M

Now we are ready to present the new algorithm SIM-M.

SIM-M$(A, B, R, \boldsymbol{f}, d_0, \epsilon, \delta_0, m, n_0)$

Input:

* A, B: $n \times n$ matrices
* R: search region in $\mathbb{C}$
* $\boldsymbol{f}$: a random vector
* d_0: precision
* ϵ: residual tolerance
* δ_0: indicator threshold
* m: size of Krylov subspace
* n_0: number of quadrature points

Output:

* generalized eigenvalues λ's inside R

1. use the center of R as the first shift and generate the associated Krylov subspaces.
2. pre-divide R into small squares of size h_0: $R_j, j = 1, \dots, J$ (these are selected squares at the initial level).

3. for $j = 1 : J$ do
   * For all quadrature points for R_j, check if the related linear systems can be solved using any one of the existing Krylov subspaces up to the given residual ϵ_0. If yes, associate R_j with that Krylov subspace. Otherwise, set the shift to be the center of R_j and construct a Krylov subspace.
4. calculate the number of the levels, denoted by K, needed to reach the precision d_0.
5. for $k = 1 : K$
   * for each selected square R_j^k at level k, check if R_j^k is resolvable.
     - if yes, compute the indicator for R_j^k and mark it when the indicator is larger than δ_0, i.e., R_j^k contains eigenvalues.
     - if R_j^k is not solvable, mark R_j^k and leave it to next level.
   * divide marked squares into four squares uniformly and move to next level.
6. post-processing the marked squares at level K, merge eigenvalues when necessary, show warnings if there exist unsolvable squares.
7. output eigenvalues (and the multiplicities).

4.3 Numerical Examples

We show some examples for SIM-M. All the test matrices are from the University of Florida Sparse Matrix Collection [44] except the last example. The computations are done using MATLAB R2017a on a MacBook Pro with 16 GB memory and a 3-GHz Intel Core i7 CPU.

4.3.1 Directed Weighted Graphs

The first group contains four non-symmetric matrices, HB/gre_115, HB/-gre_343, HB/gre_512, HB/gre_1107. These matrices represent directed weighted graphs.

Table 4.1: Comparison between SIM-M and *eig* of Example 1

	gre_115	gre_343	gre_512	gre_1107
SIM-M	3.4141s	10.2917s	14.7461s	40.2252s
eig	0.0076s	0.0918s	0.2391s	1.0947s
SIM-M/*eig*	4.5095e+02	1.1209e+02	61.6705	36.7453

We compute all eigenvalues using SIM-M and compare the results with Matlab *eig* in Table 4.1. The first row represents the four matrices. The second row contains CPU times (in seconds) used by SIM-M. The numbers in the third row are the CPU times used by Matlab *eig*. The fourth row shows the ratios by the two methods. For smaller matrices, SIM-M is much slower. However, there is a clear trend that the ratio gets smaller as the size of the matrices become larger. In Fig. 1, we show the eigenvalues computed by SIM-M and Matlab *eig*, which coincide each other.

4.3.2 Electromagnetics Problem

The second example, Bai/qc2534, is a sparse 2534×2534 matrix modeling H2+ in an electromagnetic field. The full spectrum, computed by Matlab *eig*, is shown in Fig. 2(a), in which the red rectangle is $R_1 = [-0.1, 0] \times [-0.125, 0.025]$. In Fig. 2(b), the eigenvalues are computed by SIM-M in S_1, which coincide with those computed by Matlab *eig*. The red rectangle is Fig. 2(b) is $R_2 = [-0.04, 0] \times [-0.04, 0]$. Eigenvalues in R_2 computed by SIM-M are shown in Fig. 2(c). The rectangle in Fig. 2(c) is $R_3 =$

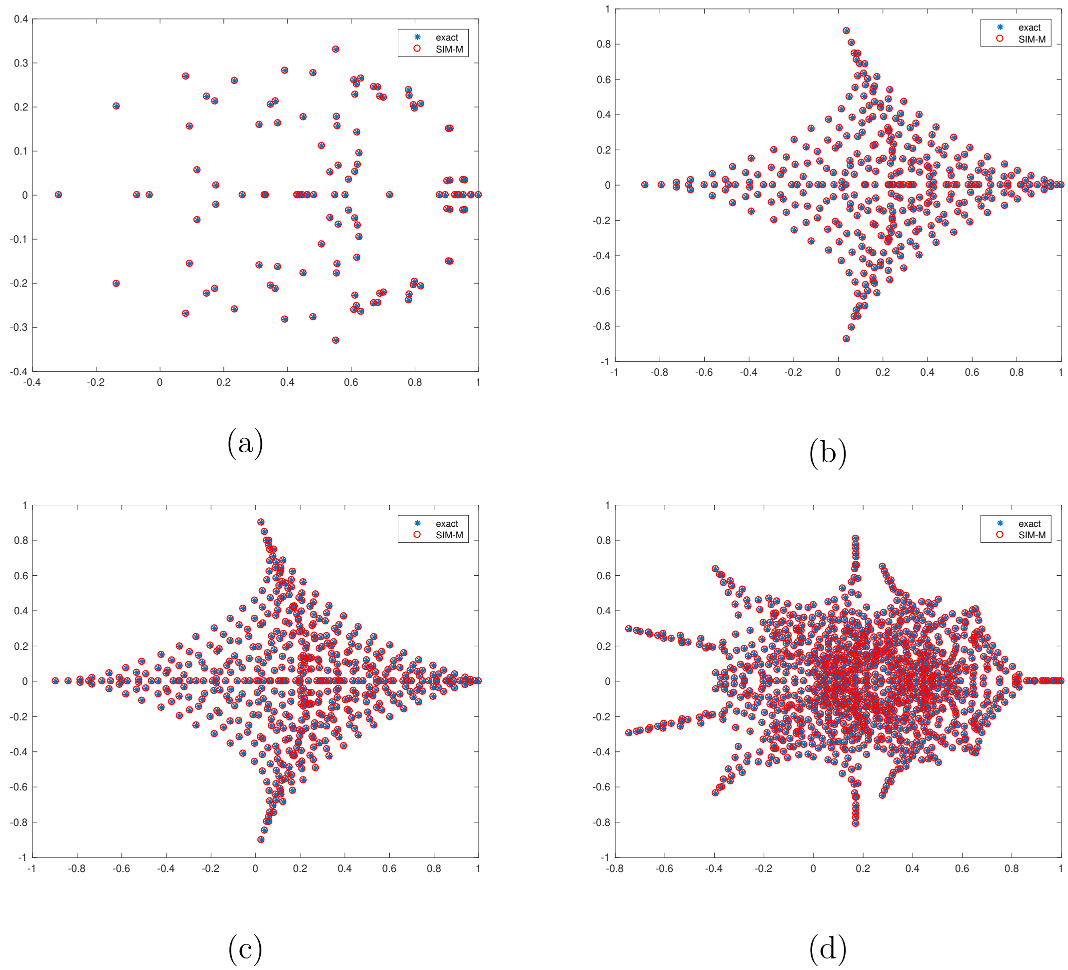


Figure 4.1: Eigenvalues computed by SIM-M and Matlab *eig* coincide. (a) HB/-gre_115. (b) HB/gre_343. (c): HB/gre_512. (d): HB/gre_1107.

$[-0.02, 0] \times [-0.03, -0.02]$. Eigenvalues in R_3 computed by SIM-M are shown in Fig. 2(d).

Table 4.2 shows the time used by Matlab *eig* to compute all eigenvalues and by SIM-M in R_1, R_2 and R_3. There are 88, 23 and 7 eigenvalues in R_1, R_2 and R_3, respectively.

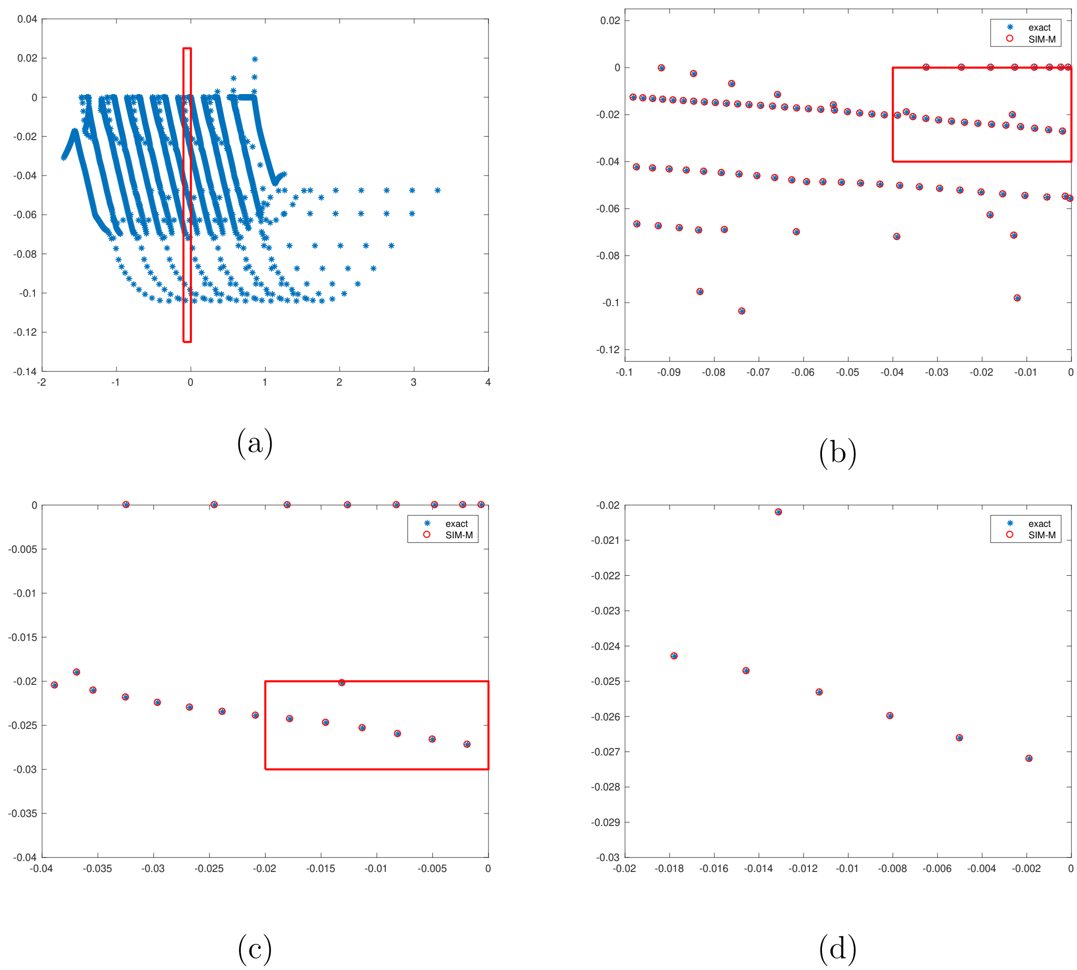


Figure 4.2: QC2534. (a): Full spectrum by Matlab *eig* (the rectangle is R_1). (b): Eigenvalues by SIM-M in R_1 (the rectangle is R_2). (c): Eigenvalues by SIM-M in R_2 (the rectangle is R_3). (d): Eigenvalues by SIM-M in R_3.

Table 4.2: Comparison between SIM-M and *eig* of Example 2

Matlab *eig*	SIM-M(R_1)	SIM-M(R_2)	SIM-M(R_3)
24.3786s	14.7445s	3.7005s	0.54645s

4.3.3 DNA Electrophoresis

The third example is a $39,082 \times 39,082$ matrix arising from DNA electrophoresis. We consider a series of nested domains

$$R_1 = [0.230, 0.270] \times [-0.0005, 0.0005],$$
$$R_2 = [0.250, 0.270] \times [-0.0005, 0.0005],$$
$$R_3 = [0.250, 0.260] \times [-0.0005, 0.0005],$$
$$R_4 = [0.254, 0.256] \times [-0.0005, 0.0005],$$

and use SIM-M to compute eigenvalues inside them. It is not possible to use Matlab *eig* to find all eigenvalues due the memory constraint. In stead, one can use Matlab *eigs* since the matrices are sparse.

In Table 4.3, time and number of eigenvalues is each domain are shown.

Table 4.3: Comparison between SIM-M and *eigs* of Example 3

	eigs(A,1000)	R_1	R_2	R_3	R_4
# of eigenvalues	1000	105	31	31	8
time	1732.4396s	588.3552s	299.4242s	214.0637s	47.8098s

Remark 4.3.0.1. Numerical results in the above two subsections indicate that a parallel version of SIM-M has the potential to be faster than the classical methods.

4.3.4 Quantum States in Disordered Media

The test matrices are sparse and symmetric arising from localized quantum states in random or disordered media [40]. The matrices A and B are of $1,966,080 \times 1,966,080$. We consider three nested domains given by

$$R_1 = [0.00, 0.60] \times [-0.05, 0.05],$$

$$R_2 = [0.00, 0.50] \times [-0.05, 0.05],$$

$$R_3 = [0.00, 0.40] \times [-0.05, 0.05],$$

In Table 4.4, time and number of eigenvalues is each domain are shown. Time of *eig*(A, B, 200) are also shown for reference.

Table 4.4: Comparison between SIM-M and *eigs* of Example 4

	eigs(A, B, 200)	R_1	R_2	R_3
# of eigenvalues	200	36	7	3
time	2469.8730s	573.1088s	112.1876s	58.9957s

Remark 4.3.0.2. The matrices are quite large for a laptop using Matlab. Matlab *eigs* are not able to handle many eigenvalues (e.g., $10,000$) due to the memory limitation and return the following message

```
>> eigs(A,B,10000,'sm')
Error using zeros
.... exceeds maximum array size preference...
```

In contrast, SIM-M uses little memory in addition to build a Krylov subspace. Consequently, one can use SIM-M to compute many eigenvalues at the cost of more time. Again, a parallel version would certain help to improve the speed.

Chapter 5

A New Fast Method of Solving the High Dimensional Elliptic Eigenvalue Problem [1]

Abstract

In this chapter, we develop a novel method to solve the elliptic PDE eigenvalue problem. The univariate multi-wavelet approach provides a simple diagonal preconditioner for second order elliptic problems, which gives an almost constant condition number for efficiently solving the corresponding linear system. Here, we shall consider a new fast numerical approach for approximating the smallest elliptic eigenvalue by using the multi-wavelet basis in the multi-grid discretization scheme. Moreover, we develop a new numerical scheme coupled with sparse grids method in the calculation. This new approach saves storage in degrees of freedom and thus is more efficient in the computation. Several numerical experiments are provided for validating the proposed numerical scheme, which show that our method

[1]This chapter has been published as an article in Applied Mathematics and Computation. https://doi.org/10.1016/j.amc.2019.03.035

retains the optimal convergence rate for the smallest eigenvalue approximation with much less computational cost comparing with 'eigs' in full grids.

Keywords: Multi-grid Discretization, Riesz Basis, Multi-wavelet, Elliptic Eigenvalue, Sparse Grids.

5.1 Introduction

The elliptic eigenvalue problem is widely used in many practical applications such as vibration models, nuclear magnetic resonance measurements, quantum mechanics and construction of heat kernels, etc [1, 45].Here we consider the high-dimensional elliptic eigenvalue problem, where standard approaches fail due to the exponentially increasing degrees of freedom w.r.t dimension d.

Algebraic methods [41, 43] fail to handle the matrix from a standard discretization even for moderate values of dimension d. Recently, several approaches have been developed to overcome this challenge. The idea is to assume that the solution could be well approximated by a low rank approximation in the tensor format. Hackbusch [46] investigated and provided the error estimate for low rank tensor approximation of elliptic eigenvalue problems in high dimension. However, as the rank often grows rapidly after each iteration, repeating the low rank truncations is needed. Kressner [47] proposed a low-rank tensor variant of locally optimal block preconditioned conjugate gradient (LOBPCG) based on hierarchical Tucker decomposition. However, specific preconditioner has to be constructed for better numerical performance.

Sparse grids method is a novel numerical approach in high dimensional approximation, which is closely related to hyperbolic crosses [48]. The main philosophy is seeking a proper truncation of the tensor product hierarchical base, which reduces the degrees of freedom from $O(N^d)$ to $O(N|\log N|^{d-1})$, where N is the number of uniform mesh in each direction. Sparse grids techniques have been integrated with finite differences [49], discontinuous Galerkin method [50], and etc. for high dimensional partial differential equations (PDEs).

The fundamental work of wavelet could be traced back to Daubechies [51]. By considering multiple generating functions, we could construct multi-wavelet with symmetry, compact support, continuity and orthogonality simultaneously [52]. Wavelet methods for PDEs have been studied for its best N-term approximation and compression properties [53, 54]. The adaptive wavelet method has been applied for Poisson's equation in high dimensionality [55]. With its well-crafted multi-wavelet, a simple diagonal preconditioner could be attained such that the preconditioned stiffness matrix has a almost constant condition number.

Two-grid discretization scheme for elliptic eigenvalue problem was first introduced by Xu and the corresponding convergence analysis for the smallest eigenvalue has been well established [56], the underlying idea is to reduce the original eigenvalue problem on the fine grid to an eigenvalue problem on a coarser grid and linear algebraic system on the fine grid. Some acceleration techniques and convergence analysis for other eigenvalues have been investigated in [57, 58]. Li [59] applied the adaptive finite element method based multi-scale discretization scheme for elliptic eigenvalue problem. Yang [60] considered shifted-inverse iteration based on the multi-grid discretizations and established the convergence for arbitrary eigenvalues under mild conditions. For our method, we adopt the two-grid discretization scheme and extend to multi-grid discretization formulation. Following the basic idea of two-grid discretization scheme, we could reduce the original eigenvalue problem on fine grid to an eigenvalue problem on a much coarser grid and linear algebraic systems on several nested finer grids. Thus, the main computational cost is solving the linear algebraic systems, we will discuss how to precondition these linear algebraic systems in details latter. Although the theoretical conclusion is only valid for the smallest eigenvalue in our method, we observe the optimal convergence rate for other eigenvalues. The theoretical analysis will be left for future study.

In this chapter, we shall consider the multi-wavelet basis coupled with sparse grids methods for approximating the second order elliptic eigenvalue problems. We shall discuss preconditioning techniques for the corresponding linear algebraic system. The rest of this paper is organized as follows: In Section 5.2 the construction of multi-wavelet and Riesz basis will be discussed. In Section 5.3 multi-grid discretization scheme with sparse grids is provided. Several numerical experiments are presented to validate the theoretical conclusions in Section 5.4.

5.2 Background

5.2.1 Univariate Orthonormal Multi-wavelet

In this subsection, we will introduce the intertwining multi-resolution analysis [52] and review the construction of univariate orthonormal multi-wavelet basis on $L^2(\mathbb{R})$, then we shall construct the univariate orthonormal multi-wavelet basis on $L^2([0,1])$ with vanishing boundary condition, where $L^2([0,1])$ refers to square-integrable function space on $[0,1]$.

A multi-resolution analysis of multiplicity r is a nested sequence of closed linear subspaces (V_p) in $L^2(\mathbb{R})$ satisfying as follows:

(a) $f \in V_p$ iff $f(2^{-p}) \in V_0$ for $p \in \mathbb{Z}$,

(b) $V_0 \subset V_1$,

(c) $\cap_{p\in\mathbb{Z}} V_p = \{0\}$,

(d) $\overline{\cup_{p\in\mathbb{Z}} V_p} = L^2(\mathbb{R})$,

(e) There are r functions $\{\phi_1, \phi_2, ..., \phi_r\}$ such that the collection $\{\phi_s(\cdot - n) | s = 1, ..., r \text{ and } n \in \mathbb{Z}\}$ is a Riesz basis for V_0.

Furthermore, if the fifth condition above could form an orthogonal basis of V_0, then we call (V_p) an orthogonal multi-resolution analysis.

Lemma 5.2.1. *If (V_p) is a multi-resolution analysis generated by compactly supported scaling functions [52], then there is some pairs of integer (q,n) and some orthogonal multiresolution analysis $(\tilde{V}_p)$ such that*

$$V_q \subset \tilde{V}_q \subset V_{q+n}.$$

We denote the generator of V_0 as $\{\phi_1, \phi_2, ..., \phi_r\}$ i.e. $V_0 = \text{span}\{\phi_j(\cdot - i) : j \in \{1, 2, \cdots, r\}, i \in \mathbb{Z}\}$, then $V_1 = \text{span}\{\phi_j(2 \cdot -i) : j \in \{1, 2, \cdots, r\}, i \in \mathbb{Z}\}$, etc. Once $(\tilde{V}_q)$ is constructed, we define the multi-wavelet subspace $\tilde{W}_l$ as the orthogonal complement of $\tilde{V}_{l-1}$ in $\tilde{V}_l$ with respect to the square-integrable inner product on $\mathbb{R}$.

Example: Piecewise linear orthonormal scaling functions.
Now we consider piecewise linear multi-wavelet, i.e. $V_0 \subset \tilde{V}_0 \subset V_1$.
Start from

$$H(x) = \begin{cases} 1 - |x| & \text{if } |x| \le 1, \\ 0 & \text{otherwise.} \end{cases}$$

Let (V_p) be the multi-resolution analysis generated by $\{\phi_1, \phi_2\}$, here $\phi_1 = \sqrt{3}H(2x)$ and $\phi_2 = \sqrt{3}H(2x - 1)$. Following from [52], we shall construct the scaling functions $\tilde{\Phi} = (\tilde{\phi}_1, \tilde{\phi}_2, \tilde{\phi}_3)$ shown in Figure 5.1. Figure 5.2 plots the corresponding wavelet functions.

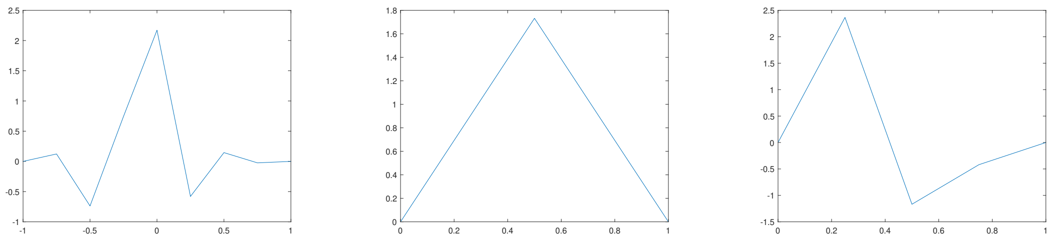


Figure 5.1: Piecewise-linear orthonormal scaling functions $\tilde{\phi}_1$, $\tilde{\phi}_2$, $\tilde{\phi}_3$.

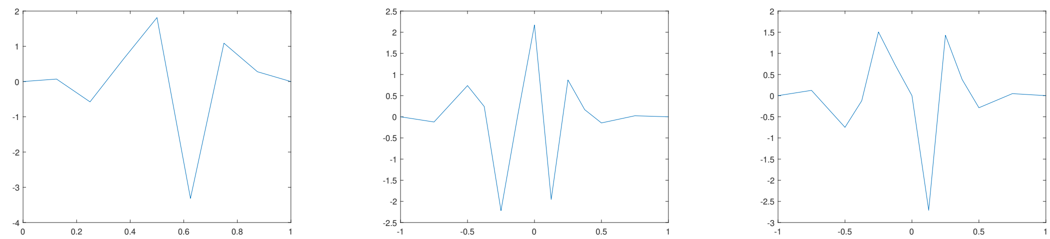


Figure 5.2: Piecewise-linear orthonormal wavelets $\tilde{\psi}_1$, $\tilde{\psi}_2$, $\tilde{\psi}_3$.

The collection $\{\phi_{j,l,i} := 2^{l/2}\tilde{\phi}_j(2^l \cdot -i) : j \in \{1, 2, \cdots, k\}, i \in \mathbb{Z}\}$ and $\{\psi_{j,l,i} := 2^{(l-1)/2}\tilde{\psi}_j(2^{(l-1)} \cdot -i) : j \in \{1, 2, \cdots, k\}, i \in \mathbb{Z}\}$ are orthonormal bases for $\tilde{V}_l$ and $\tilde{W}_l$, i.e. the piecewise linear multi-wavelet introduced above with $k = 3$. Thus we have the hierarchical decomposition

$$\begin{aligned}
\tilde{V}_{l_k} &= \tilde{V}_{l_0} \oplus \tilde{W}_{l_0+1} \oplus \tilde{W}_{l_0+2} \cdots \oplus \tilde{W}_{l_k} \qquad l_0 \geq 0, \\
L_2(\mathbb{R}) &= \tilde{V}_{l_0} \oplus \tilde{W}_{l_0+1} \oplus \tilde{W}_{l_0+2} \cdots \qquad l_0 \geq 0.
\end{aligned}$$

However, the construction above is for $L_2(\mathbb{R})$. As for multi-wavelet on $L_2([0,1])$ with vanishing boundary condition, the construction is nontrivial in general [52]. Fortunately, we still retain multi-wavelet basis on $L_2([0,1])$ with vanishing boundary condition by restricting multi-wavelet basis of $L_2(\mathbb{R})$ on $[0,1]$ for certain types of multi-wavelet, i.e. the piecewise linear multi-wavelet.

Now we shall define the multi-wavelet basis on $L_2([0,1])$ as following

$$\Psi := \{\psi_\lambda : \lambda \in \Lambda\} = \{\phi_{j,1,i}|_{[0,1]}\} \bigcup (\bigcup_{l>1} \{\psi_{j,l,i}|_{[0,1]}\}).$$

So Ψ is the union of $\tilde{V}_1|_{[0,1]}$ and $\tilde{W}_l|_{[0,1]}$ for all $l > 1$ indexed by λ. λ is the triple indexes (j, l, i) and Λ is collection of all valid triples (j, l, i). Here we use $|\lambda|$ to denote l, the level information of multi-wavelet basis.

5.2.2 Riesz Basis in Energy Norm

We shall consider the following elliptic eigenvalue problem in d (d=1, 2, 3) dimension of $\Omega = (0,1)^d$:

$$-\Delta u = \lambda u, \text{ in } \Omega, \tag{5.2.1}$$

$$u = 0, \text{ on } \partial\Omega. \tag{5.2.2}$$

Define the bilinear form as follows

$$a(u,v) := \int_\Omega \nabla u \cdot \nabla v dx \quad (v \in H_0^1(\Omega)). \tag{5.2.3}$$

We use $H_0^1(\Omega)$ to denote Sobolev function space with vanishing trace.

Next we will demonstrate some properties with the basis above, namely, the Riesz basis in energy norm.

Lemma 5.2.2. *The bilinear form $a(u,v)$ is symmetric positive definite and elliptic in the sense that*

$$||v||_a^2 := a(v,v) \sim ||v||_H^2 \quad v \in H_0^1(\Omega). \tag{5.2.4}$$

Here $\sim$ means that both quantities can be uniformly bounded by constant multiples of each other and $||\cdot||_H$ is the corresponding Sobolev norm. Now we consider the one dimensional case, the multi-wavelet basis defined in last section satisfies Jackson and Bernstein estimates [61], thus we have the following estimation

$$||v||_a^2 \sim ||2^{|\lambda|}\langle v, \psi_\lambda\rangle_{L_2(0,1)}||^2_{l_2(\lambda\in\Lambda)} \quad \forall v \in H_0^1(0,1). \tag{5.2.5}$$

Define an infinite diagonal matrix $\boldsymbol{D}^{\pm}$ as following

$$(\boldsymbol{D}^{\pm})_{\lambda,\tilde{\lambda}} := 2^{\pm|\lambda|}\delta_{(\lambda,\tilde{\lambda})}. \tag{5.2.6}$$

Where $\delta_{(\lambda,\tilde{\lambda})}$ is the Kronecker Delta function.

Also we denote $\boldsymbol{v} := \langle v, \psi_\lambda\rangle_{L_2(0,1)}$ as an infinite vector indexed by $\lambda \in \Lambda$. Then (5.2.5) is equivalent to

$$||v||_a = ||\boldsymbol{v}^T\Psi||_a \sim ||\boldsymbol{D}^+\boldsymbol{v}||_{l_2}, \tag{5.2.7}$$

Next, we shall scale the multi-wavelet basis Ψ as following

$$||\boldsymbol{v}^T\Psi||_a = ||(\boldsymbol{D}^+\boldsymbol{v})^T\boldsymbol{D}^-\Psi||_a \sim ||\boldsymbol{D}^+\boldsymbol{v}||_{l_2}, \tag{5.2.8}$$

which means there exists $0 < c \leq C$ such that

$$c||\boldsymbol{D}^+\boldsymbol{v}||_{l_2} \leq ||(\boldsymbol{D}^+\boldsymbol{v})^T\boldsymbol{D}^-\Psi||_a \leq C||\boldsymbol{D}^+\boldsymbol{v}||_{l_2}. \tag{5.2.9}$$

Theorem 5.2.3. *Given the Riesz basis $\boldsymbol{D}^-\Psi$ for the energy norm $||\cdot||_a$, the condition number $\kappa_{\boldsymbol{D}^-\Psi}$ of the Gramian matrix $(\boldsymbol{D}^-\phi_\lambda, \boldsymbol{D}^-\phi_{\tilde{\lambda}}) := (\boldsymbol{D}^-\phi_\lambda, \boldsymbol{D}^-\phi_{\tilde{\lambda}})_a$:*

$$\kappa_{\boldsymbol{D}^-\Psi} \leq \left(\frac{C}{c}\right)^2 \tag{5.2.10}$$

where c and C are constants defined in (5.2.9).

The Gramian matrix $(\boldsymbol{D}^-\phi_\lambda, \boldsymbol{D}^-\phi_{\tilde{\lambda}})$ is equal to $\boldsymbol{D}^-(\psi_\lambda, \psi_{\tilde{\lambda}})_a\boldsymbol{D}^-$, which shows the condition number of the finite discretized bilinear form $(\psi_\lambda, \psi_{\tilde{\lambda}})_a$ preconditioned by the application of $\boldsymbol{D}^-$ of (5.2.6) is indeed uniformly bounded. Furthermore, $\boldsymbol{D}^-$ can be simplified by just computing the diagonal entries of $(\psi_\lambda, \psi_{\tilde{\lambda}})_a$ since by applying (5.2.5) we have as following

$$||\psi_\lambda||_a^2 \sim ||2^{|\lambda|}\langle \psi_\lambda, \psi_{\tilde{\lambda}}\rangle_{L_2(0,1)}||^2_{l_2(\tilde{\lambda}\in\Lambda)} = 2^{2|\lambda|} \tag{5.2.11}$$

$$(\psi_\lambda, \psi_{\tilde{\lambda}}\delta_{(\lambda,\tilde{\lambda})})_a^{\frac{-1}{2}} \sim \boldsymbol{D}^- \tag{5.2.12}$$

Thus we shall use the diagonal of the stiffness matrix to construct the preconditioner $\boldsymbol{D}^-$.

We will demonstrate the condition number of the stiffness matrix and the uniform boundedness of preconditioned stiffness matrix with preconditioning in (5.2.12). Here DOFs denotes degrees of freedom and CN denotes condition number. For the experiment, we consider $d = 1$ and take linear multi-wavelet basis as example. Tables 5.1-5.2 compare the condition number of original stiffness matrix and preconditioned stiffness matrix. From Table 5.2, one can observe that the condition number of the preconditioned stiffness matrix is almost a constant which agrees with our theoretical results, while the condition number of the original problem shown in Table 5.1 increases by 4.

Level=$\|\lambda\|$	3	4	5	6	7	8	9	10
DOFs	47	95	191	383	767	1535	3071	6143
CN	5.2E3	2.1E4	8.4E4	3.4E5	1.4E6	5.4E6	2.2E7	8.6E7

Table 5.1: Condition number of 1D stiffness matrix (linear multi-wavelet).

Level=$\|\lambda\|$	3	4	5	6	7	8	9	10
DOFs	47	95	191	383	767	1535	3071	6143
CN	28.85	29.92	30.64	31.24	31.68	32.05	32.33	32.57

Table 5.2: Condition number of 1D preconditioned stiffness matrix (linear multi-wavelet).

Next we consider the arbitrary d dimensional case and its Riesz basis on

$H^1_0(\Omega)$. Similarly, we define the tensor product basis

$$\hat{\mathbf{\Psi}} := \{\boldsymbol{\psi}_{\boldsymbol{\lambda}} := \otimes_{m=1}^{d} \psi_{\lambda_m} : \boldsymbol{\lambda} \in \mathbf{\Lambda} := \prod_{m=1}^{d} \Lambda_m\}, \tag{5.2.13}$$

and $\boldsymbol{D}_d^-$ is defined similar to (5.2.12). $\hat{\mathbf{\Psi}}$ is the orthonormal basis for $L_2(\Omega)$ and $\boldsymbol{D}_d^- \hat{\mathbf{\Psi}}$ is Riesz basis for $H^1_0(\Omega)$. $\boldsymbol{D}_{\boldsymbol{d}}^- a(\boldsymbol{\psi}_{\boldsymbol{\lambda}}, \boldsymbol{\psi}_{\tilde{\boldsymbol{\lambda}}}) \boldsymbol{D}_{\boldsymbol{d}}^-$ shares the same condition number with $\boldsymbol{D}^- a(\psi_\lambda, \psi_{\tilde{\lambda}}) \boldsymbol{D}^-$ in one dimension [55].

Remark 5.2.3.1. The significance of orthonormality in Ψ lays the foundation that $\boldsymbol{D}_{\boldsymbol{d}}^- a(\boldsymbol{\psi}_{\boldsymbol{\lambda}}, \boldsymbol{\psi}_{\tilde{\boldsymbol{\lambda}}}) \boldsymbol{D}_{\boldsymbol{d}}^-$ has uniform condition w.r.t dimension d. If we choose non-orthonormal multi-wavelet basis, condition number of the preconditioned $\boldsymbol{D}_{\boldsymbol{d}}^- a(\boldsymbol{\psi}_{\boldsymbol{\lambda}}, \boldsymbol{\psi}_{\tilde{\boldsymbol{\lambda}}}) \boldsymbol{D}_{\boldsymbol{d}}^-$ will grows exponentially w.r.t dimension d [55].

5.2.3 Sparse Grids for Multi-wavelet Basis

In this subsection, we shall introduce multi-wavelet basis for full grids and sparse grids in d dimensionality. $\hat{\mathbf{\Psi}}$ is defined as the tensor product of the one-dimensional hierarchical decomposition [62]. We consider the finite element space for computation purpose. Define the full grids (FG) $\hat{\mathbf{\Psi}}_l^f$ of level l

$$\hat{\mathbf{\Psi}}_l^f := \{\boldsymbol{\psi}_{\boldsymbol{\lambda}} \in \hat{\mathbf{\Psi}} : ||\boldsymbol{\lambda}||_{l_\infty} <= l\},$$

Here $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, ..., \lambda_d)$, $||\boldsymbol{\lambda}||_{l_\infty} = \max_{i=1}^d |\lambda_i|$. By contrast, the sparse grids, which is based on a selection of full tensor product of hierarchical basis, could significantly reduce degrees of freedom, while keeps almost the same accuracy.

Similar to full grids, we define the sparse grids (SG) $\hat{\mathbf{\Psi}}_l^s$ of level l

$$\hat{\mathbf{\Psi}}_l^s := \{\boldsymbol{\psi}_{\boldsymbol{\lambda}} \in \hat{\mathbf{\Psi}} : ||\boldsymbol{\lambda}||_{l_1} <= l\}.$$

Here we denote $||\boldsymbol{\lambda}||_{l_1} = \sum_{i=1}^d |\lambda_i|$. In fact, with sparse grids method the degree of freedoms could be reduced from order $O(h^{-d})$ to $O(h^{-1}|\log_2 h|^{d-1})$

for d-dimensional problems [50], where h is the size of uniform mesh in each dimension.

Level	3	4	5	6	7	8
DOFs	2209	9025	36481	146689	588289	2356225
Condition Number	28.8	29.9	30.6	31.3	-	-

Table 5.3: Condition number of 2D FG preconditioned stiffness matrix (linear multi-wavelet)

Level	3	4	5	6	7	8
DOFs	625	1537	3649	8449	19201	43009
Condition Number	28.7	29.8	30.5	31.1	31.6	31.9

Table 5.4: Condition number of 2D SG preconditioned stiffness matrix (linear multi-wavelet)

Tables 5.3-5.4 report the condition number from full grids (FG) and sparse grids (SG) for two dimensional problem after applying diagonal preconditioning. The condition numbers of preconditioned stiffness matrix for two dimensional problem are the same as the one dimensional case shown in Table 5.2. Furthermore, the savings of DOFs in SG methods are significant compared with FG methods.

Remark 5.2.3.2. We denote "-" for the limitation of computational resource in Table 5.3.

5.3 Multi-Grid Discretization Scheme with Sparse Grids

In this section, we adopt two-grid discretization scheme and refer more details in [56]. We propose the multi-grid discretization scheme with multi-wavelet basis in sparse grids for approximating the smallest eigenvalue. The algorithm is summarized as following:

(a) Solve an eigenvalue problem on an initial coarse grid: choose an initial level l_0, find the smallest eigenvalue λ_{l_0} and $u_{l_0} \in \hat{\mathbf{\Psi}}^s_{l_0}$ such that $||\nabla u_{l_0}||_{L_2} = 1$, and,

$$\int_\Omega \nabla u_{l_0} \cdot \nabla v dx = \lambda_{l_0}(u_{l_0}, v) \quad (\forall v \in \hat{\mathbf{\Psi}}^s_{l_0}).$$

And $i \Leftarrow 0$.

(b) Let $i \Leftarrow i + 1$, solve a linear algebraic system on a finer grid: find $u_{l_i} \in \hat{\mathbf{\Psi}}^s_{l_i}$ such that

$$\int_\Omega \nabla u_{l_i} \cdot \nabla v dx = \lambda_{l_{i-1}}(u_{l_{i-1}}, v) \quad (\forall v \in \hat{\mathbf{\Psi}}^s_{l_i}), \tag{5.3.14}$$

and compute the Rayleigh quotient

$$\lambda_{l_i} = \frac{||\nabla u_{l_i}||^2_{L_2}}{||u_{l_i}||^2_{L_2}}.$$

(c) If $i \leq Max$ then goes to 2, else stops. Here Max denotes the max level information.

The eigenpair $(\lambda_{l_{Max}}, u_{l_{Max}})$ is approximation of original problem in $\hat{\mathbf{\Psi}}^s_{l_{Max}}$. Our contribution is to utilize (5.3.14) with fast method. We have the following algebraic systems after discretizing (5.3.14)

$$A_{l_i} x_{l_i} = f_{l_{i-1}}. \tag{5.3.15}$$

A diagonal preconditioner D_{l_i} by (5.2.12) could be easily constructed as $D_{l_i} = (\text{diag}(A_{l_i}))^{-1/2}$ such that the condition number of $D_{l_i}A_{l_i}D_{l_i}$ is uniformly bounded as we demonstrate in Tables 5.2-5.4.

Multiply D_{l_i} to both sides of (5.3.15), and rewrite the equation as

$$D_{l_i}A_{l_i}D_{l_i}D_{l_i}^{-1}x_{l_i} = D_{l_i}f_{l_{i-1}}. \tag{5.3.16}$$

Denote $\tilde{A}_{l_i} = D_{l_i}A_{l_i}D_{l_i}$, $\tilde{x}_{l_i} = D_{l_i}^{-1}x_{l_i}$ and $\tilde{f}_{l_{i-1}} = D_{l_i}f_{l_{i-1}}$, then (5.3.16) is equivalent to

$$\tilde{A}_{l_i}\tilde{x}_{l_i} = \tilde{f}_{l_{i-1}}. \tag{5.3.17}$$

After we solve (5.3.17) with iterative methods such as generalized minimal residual method (GMRES), we retain the solution of (5.3.15).

5.4 Numerical Experiments

In this section, we provide multi-dimensional numerical results to demonstrate the performance of our method. Numerical experiments have been carried out with linear and cubic finite elements. According to theoretical conclusions, we expect the convergence rates as $\mathcal{O}(h^{2p})$ for the smallest eigenvalue, where p is the the order of basis. The mesh size is denoted as $h = \dfrac{1}{2^l}$, where l is the level information of finite multi-wavelet basis. All the calculations are performed with MATLAB R2017a on Dell Workstation equipped with Windows 10 system, two Intel Quad-Core Xeon X5697 3.59 GHz CPUs and 48 GB of main memory.

5.4.1 Two Dimensional Test

We solve the following two-dimensional problem with constant coefficient on $\Omega = (0,1)^2$

$$-\Delta u \;=\; \lambda u, \text{ in } \Omega,$$

$$u = 0, \text{ on } \partial\Omega.$$

The exact eigenvalues are given by $\lambda = \pi^2(k_x^2 + k_y^2)$ and the eigenfunctions are $\sin(k_x \pi x)\sin(k_y \pi y)$ with $k_x = 1, 2, 3, \ldots$ and $k_y = 1, 2, 3, \ldots$. We test our method and compare our results with the conventional method, namely, full grids methods using "eigs" command in Matlab. We compare several numerical performance like convergence rate, computational cost (in seconds) and degrees of freedom (DOFs).

The comparison between the full grids method and our proposed method with piecewise linear basis is presented in Tables 5.5-5.6. The reduction of DOFs due to sparse grids is dramatic. Furthermore, the diagonal preconditioning technique plays another significant role for speeding up. Almost constant ratio between computational cost and degrees of freedom is achieved in Table 5.6. The new method also retains optimal convergence order for the smallest eigenvalue, which is second order for piecewise linear basis.

Level	4	5	6	7	8	9
DOFs	4417	11137	26881	62977	144385	325633
Cost	0.29	1.81	11.2	59.4	402.9	-
Error	2.31e-3	5.78e-4	1.45e-4	3.61e-5	9.03e-6	-
Conv.Rate		2.00	2.00	2.00	1.99	-

Table 5.5: 2D SG method using "eigs" for first eigenvalue (linear multi-wavelet)

Furthermore, we conduct experiments for the other eigenvalues' approximation in Tables 5.7-5.8 and we observe the optimal convergence rates by using both linear and cubic finite elements.

Remark 5.4.0.1. The convergence rate drops when the error is less than 10^{-10}, one reason for this phenomenon is that the iterative method in

Level	4	5	6	7	8	9
DOFs	4417	11137	26881	62977	144385	325633
Cost		0.23	0.63	1.47	4.18	10.53
Error	2.31e-3	5.78e-4	1.45e-4	3.62e-5	9.09e-6	2.31e-6
Conv.Rate		2.00	2.00	2.00	1.99	1.98

Table 5.6: 2D SG method with multi-grid scheme for first eigenvalue (linear multi-wavelet)

Level	4	5	6	7	8	9
DOFs	4417	11137	26881	62977	144385	325633
Cost		0.21	0.57	1.39	4.04	10.54
Error	9.48e-2	2.37e-2	5.92e-3	1.48e-3	3.71e-4	9.38e-5
Conv.Rate		2.00	2.00	2.00	2.00	1.98

Table 5.7: 2D SG method with multi-grid scheme for 5th eigenvalue (linear multi-wavelet)

Level	3	4	5	6	7
DOFs	1153	2817	6657	15361	34817
Cost		0.08	0.19	0.44	1.1
Error	4.98e-3	1.19e-4	1.35e-6	1.91e-8	3.27e-10
Conv.Rate		5.37	6.47	6.14	5.87

Table 5.8: 2D SG method with multi-grid scheme for 20th eigenvalue (cubic multi-wavelet)

MATLAB like GMRES may fail to converge if the tolerance is too small.

5.4.2 Three Dimensional Test

Next we solve three-dimensional problem with constant coefficient on $\Omega = (0,1)^3$

$$
\begin{aligned}
-\Delta u &= \lambda u, \text{ in } \Omega, \\
u &= 0, \text{ on } \partial\Omega.
\end{aligned}
$$

The exact eigenvalues are given by $\lambda = \pi^2(k_x^2 + k_y^2 + k_z^2)$ and the eigenfunctions are $\sin(k_x \pi x)\sin(k_y \pi y)\sin(k_z \pi z)$ with $k_x, k_y, k_z = 1, 2, 3 \ldots$.

We perform numerical experiment to compute the smallest eigenvalue in piecewise linear basis and result is presented in Table 5.9.

Level	2	3	4	5	6	7
1DOFs	2015	6191	17567	47231	122111	306431
Cost		0.32	0.26	0.69	1.99	6.28
Error	5.57e-2	1.39e-2	3.48e-3	8.70e-4	2.18e-4	5.49e-5
Conv.Rate		2.00	2.00	2.00	2.00	1.99

Table 5.9: 3D SG method with multi-grid scheme for first eigenvalue (linear multiwavelet)

Similarly, we compute non-first eigenvalues in both linear and cubic basis in Tables 5.10-5.11.

Level	2	3	4	5	6	7
DOFs	2015	6191	17567	47231	122111	306431
Cost		0.10	0.22	0.61	1.80	5.82
Error	6.33e-1	1.58e-1	3.95e-2	9.87e-3	2.47e-3	6.16e-4
Conv.Rate		2.00	2.00	2.00	2.00	1.99

Table 5.10: 3D SG method with multi-grid scheme for 5th eigenvalue (linear multi-wavelet)

Level	2	3	4	5	6
DOFs	5215	15807	44415	118527	304639
Cost		0.32	1.05	3.15	9.94
Error	1.57e-4	2.57e-6	3.93e-8	6.16e-10	1.36e-11
Conv.Rate		5.93	6.03	6.00	5.50

Table 5.11: 3D SG method with multi-grid scheme of 5th eigenvalue (cubic multi-wavelet)

5.4.3 2D L-shaped Domain Test

Finally, we solve two-dimensional problem with constant coefficient on $\Omega = (0,1)^2 \backslash (1/2,1) \times (0,1/2)$

$$\begin{aligned} -\Delta u &= \lambda u, \text{ in } \Omega, \\ u &= 0, \text{ on } \partial\Omega. \end{aligned}$$

For the L-shaped domain, the first eigenvalue can not be obtained exactly. To study the convergence rate, we use the relative error calculated with reference eigenvalue. We study the convergence rate of the first eigenvalue. As we know, the first eigenfunction only has $H^{5/3}$ regularity [1]. In Table 5.12, we observe the superconvergence as second order. However, at the finest level $l = 10$, the first eigenvalue is 38.69 for our method while the reference eigenvalue is around 38.56 by finite element method [1, p. 71]. In Figure 5.3, we plot first eigenfunction for this test.

Level	5	6	7	8	9	10
DOFs	8086	19603	46095	105994	239620	534525
Cost(s)	0.21	0.44	1.16	2.76	7.74	19.21
Relative Error	3.45e-4	8.93e-5	2.26e-5	5.72e-6	1.45e-6	3.67e-7
Conv.Rate		1.95	1.98	1.98	1.98	1.98

Table 5.12: 2D SG method with multi-grid scheme for L-shape problem (linear multi-wavelet)

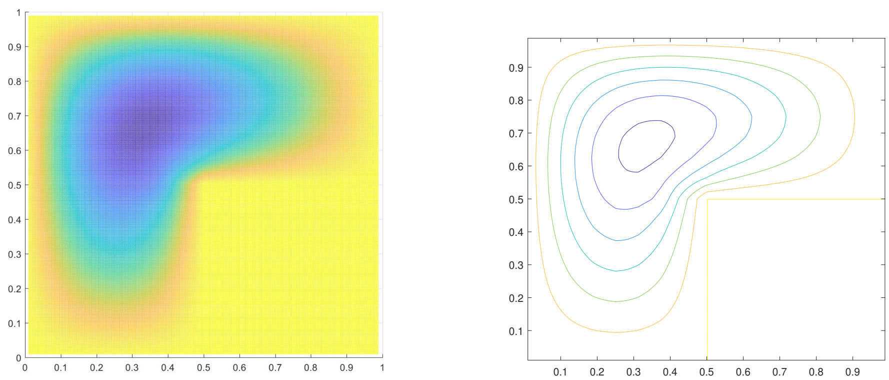


Figure 5.3: First eigenfunction of L-shaped domain

The phenomenon of superconvergence will be left for further study. Furthermore, in [63], non-overlapping domain decomposition method is proposed to solve Poisson problem with irregular domain, we will consider this technique for elliptic eigenvalue problem with more general domain in the future.

5.5 Conclusion

In this chapter, we develop a novel method for high dimensional elliptic eigenvalue problem on the tensorized domain. The theoretical convergence rate for the smallest eigenvalue is verified by the numerical experiments and our method achieves almost the same order in accuracy as the "eigs" with far less time due to the integration of multi-grid discretization scheme, well-constructed multi-wavelet basis, and sparse grids techniques. Furthermore, we observe the empirical convergence for other eigenvalues, which will be left for future investigation.

References

[1] Jiguang Sun and Aihui Zhou. *Finite element methods for eigenvalue problems*. Chapman and Hall/CRC, 2016.

[2] Xia Ji, Jiguang Sun, and Tiara Turner. Algorithm 922: a mixed finite element method for helmholtz transmission eigenvalues. *ACM Transactions on Mathematical Software (TOMS)*, 38(4):29, 2012.

[3] Yousef Saad. *Numerical methods for large eigenvalue problems: revised edition*, volume 66. Siam, 2011.

[4] Eric Polizzi. Density-matrix-based algorithm for solving eigenvalue problems. *Physical Review B*, 79(11):115112, 2009.

[5] Ken Goldberg, Theresa Roeder, Dhruv Gupta, and Chris Perkins. Eigentaste: A constant time collaborative filtering algorithm. *information retrieval*, 4(2):133–151, 2001.

[6] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[7] David L Colton, Rainer Kress, and Rainer Kress. *Inverse acoustic and electromagnetic scattering theory*, volume 93. Springer, 1998.

[8] Fioralba Cakoni, David Colton, Peter Monk, and Jiguang Sun. The inverse electromagnetic scattering problem for anisotropic media. *Inverse Problems*, 26(7):074004, 2010.

[9] Jiguang Sun. Iterative methods for transmission eigenvalues. *SIAM Journal on Numerical Analysis*, 49(5):1860–1874, 2011.

[10] Fioralba Cakoni and Houssem Haddar. Transmission eigenvalues in inverse scattering theory, 2012.

[11] David Colton, Peter Monk, and Jiguang Sun. Analytical and computational methods for transmission eigenvalues. *Inverse Problems*, 26(4):045011, 2010.

[12] Xinming Wu and Wenbin Chen. Error estimates of the finite element method for interior transmission problems. *Journal of Scientific Computing*, 57(2):331–348, 2013.

[13] Jiguang Sun and Liwei Xu. Computation of maxwell's transmission eigenvalues and its applications in inverse medium problems. *Inverse Problems*, 29(10):104013, 2013.

[14] Jing An and Jie Shen. A spectral-element method for transmission eigenvalue problems. *Journal of Scientific Computing*, 57(3):670–688, 2013.

[15] Andreas Kleefeld. A numerical method to compute interior transmission eigenvalues. *Inverse Problems*, 29(10):104012, 2013.

[16] Fioralba Cakoni, Peter Monk, and Jiguang Sun. Error analysis for the finite element approximation of transmission eigenvalues. *Computational Methods in Applied Mathematics*, 14(4):419–427, 2014.

[17] Tiexiang Li, Wei-Qiang Huang, Wen-Wei Lin, and Jijun Liu. On spectral analysis and a novel algorithm for transmission eigenvalue problems. *Journal of Scientific Computing*, 64(1):83–108, 2015.

[18] Yidu Yang, Hai Bi, Hao Li, and Jiayu Han. Mixed methods for the helmholtz transmission eigenvalues. *SIAM Journal on Scientific Computing*, 38(3):A1383–A1403, 2016.

[19] John E Osborn. Spectral approximation for compact operators. *Mathematics of computation*, 29(131):712–725, 1975.

[20] Anne Cossonnière, Houssem Haddar, et al. Surface integral formulation of the interior transmission problem. *Journal of Integral Equations and Applications*, 25(3):341–376, 2013.

[21] Wolf-Jürgen Beyn. An integral method for solving nonlinear eigenvalue problems. *Linear Algebra and its Applications*, 436(10):3839–3863, 2012.

[22] Jiguang Sun. An eigenvalue method using multiple frequency data for inverse scattering problems. *Inverse Problems*, 28(2):025012, 2012.

[23] Armin Lechleiter and Marcel Rennoch. Inside-outside duality and the determination of electromagnetic interior transmission eigenvalues. *SIAM Journal on Mathematical Analysis*, 47(1):684–705, 2015.

[24] Drossos Gintides and Nikolaos Pallikarakis. A computational method for the inverse transmission eigenvalue problem. *Inverse Problems*, 29(10):104010, 2013.

[25] Xia Ji and Jiguang Sun. A multi-level method for transmission eigenvalues of anisotropic media. *Journal of Computational Physics*, 255:422–435, 2013.

[26] Xia Ji, Jiguang Sun, and Hehu Xie. A multigrid method for helmholtz transmission eigenvalue problems. *Journal of Scientific Computing*, 60(2):276–294, 2014.

[27] George C Hsiao, Fengshan Liu, Jiguang Sun, and Liwei Xu. A coupled bem and fem for the interior transmission problem in acoustics. *Journal of computational and applied mathematics*, 235(17):5213–5221, 2011.

[28] Tosio Kato. *Perturbation theory for linear operators*, volume 132. Springer Science & Business Media, 2013.

[29] Tetsuya Sakurai and Hiroshi Sugiura. A projection method for generalized eigenvalue problems using numerical integration. *Journal of computational and applied mathematics*, 159(1):119–128, 2003.

[30] Guojian Yin, Raymond H Chan, and Man-Chung Yeung. A feast algorithm with oblique projection for generalized eigenvalue problems. *Numerical Linear Algebra with Applications*, 24(4):e2092, 2017.

[31] Stefan Goedecker. Linear scaling electronic structure methods. *Reviews of Modern Physics*, 71(4):1085, 1999.

[32] Anthony P Austin, Peter Kravanja, and Lloyd N Trefethen. Numerical algorithms based on analytic function values at roots of unity. *SIAM Journal on Numerical Analysis*, 52(4):1795–1821, 2014.

[33] Gene H Golub and Henk A Van der Vorst. Eigenvalue computation in the 20th century. *Journal of Computational and Applied Mathematics*, 123(1-2):35–65, 2000.

[34] Stefano Boccaletti, Vito Latora, Yamir Moreno, Martin Chavez, and D-U Hwang. Complex networks: Structure and dynamics. *Physics reports*, 424(4-5):175–308, 2006.

[35] Ruihao Huang, Allan A Struthers, Jiguang Sun, and Ruming Zhang. Recursive integral method for transmission eigenvalues. *Journal of Computational Physics*, 327:830–840, 2016.

[36] Karl Meerbergen. The solution of parametrized symmetric linear systems. *SIAM journal on matrix analysis and applications*, 24(4):1038–1059, 2003.

[37] Yousef Saad. *Iterative methods for sparse linear systems*, volume 82. siam, 2003.

[38] Philip J Davis and Philip Rabinowitz. *Methods of numerical integration*. Courier Corporation, 2007.

[39] Richard B Lehoucq, Danny C Sorensen, and Chao Yang. *ARPACK users' guide: solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods*, volume 6. Siam, 1998.

[40] Douglas N Arnold, Guy David, David Jerison, Svitlana Mayboroda, and Marcel Filoche. Effective confining potential of quantum states in disordered media. *Physical review letters*, 116(5):056602, 2016.

[41] Zhaojun Bai, James Demmel, Jack Dongarra, Axel Ruhe, and Henk van der Vorst. *Templates for the solution of algebraic eigenvalue problems: a practical guide*. SIAM, 2000.

[42] Ping Tak Peter Tang and Eric Polizzi. Feast as a subspace iteration eigensolver accelerated by approximate spectral projection. *SIAM Journal on Matrix Analysis and Applications*, 35(2):354–390, 2014.

[43] Ru Huang, Jiguang Sun, and C Yang. Recursive integral method with cayley transformation. *Numerical Linear Algebra with Applications*, 25(6):e2199, 2018.

[44] Timothy A Davis and Yifan Hu. The university of florida sparse matrix collection. *ACM Transactions on Mathematical Software (TOMS)*, 38(1):1, 2011.

[45] Denis S Grebenkov and B-T Nguyen. Geometrical structure of laplacian eigenfunctions. *siam REVIEW*, 55(4):601–667, 2013.

[46] Wolfgang Hackbusch, Boris N Khoromskij, Stefan Sauter, and Eugene E Tyrtyshnikov. Use of tensor formats in elliptic eigenvalue problems. *Numerical Linear Algebra with Applications*, 19(1):133–151, 2012.

[47] Daniel Kressner and Christine Tobler. Preconditioned low-rank methods for high-dimensional elliptic pde eigenvalue problems. *Computational Methods in Applied Mathematics Comput. Methods Appl. Math.*, 11(3):363–381, 2011.

[48] Vladimir Nikolaevich Temlyakov. Approximations of functions with bounded mixed derivative. *Trudy Matematicheskogo Instituta imeni VA Steklova*, 178:3–113, 1986.

[49] Michael Griebel. Adaptive sparse grid multilevel methods for elliptic pdes based on finite differences. *Computing*, 61(2):151–179, 1998.

[50] Zixuan Wang, Qi Tang, Wei Guo, and Yingda Cheng. Sparse grid discontinuous galerkin methods for high-dimensional elliptic equations. *Journal of Computational Physics*, 314:244–263, 2016.

[51] Ingrid Daubechies. *Ten lectures on wavelets*, volume 61. Siam, 1992.

[52] George C Donovan, Jeffrey S Geronimo, and Douglas P Hardin. Orthogonal polynomials and the construction of piecewise polynomial smooth wavelets. *SIAM Journal on Mathematical analysis*, 30(5):1029–1056, 1999.

[53] Albert Cohen and Roland Masson. Wavelet methods for second-order elliptic problems, preconditioning, and adaptivity. *SIAM Journal on Scientific Computing*, 21(3):1006–1026, 1999.

[54] Albert Cohen, Wolfgang Dahmen, and Ronald DeVore. Adaptive wavelet methods for elliptic operator equations: convergence rates. *Mathematics of Computation*, 70(233):27–75, 2001.

[55] Tammo Jan Dijkema, Christoph Schwab, and Rob Stevenson. An adaptive wavelet method for solving high-dimensional elliptic pdes. *Constructive approximation*, 30(3):423, 2009.

[56] Jinchao Xu and Aihui Zhou. A two-grid discretization scheme for eigenvalue problems. *Mathematics of Computation*, 70(233):17–25, 2001.

[57] Xiaozhe Hu and Xiaoliang Cheng. Acceleration of a two-grid method for eigenvalue problems. *Mathematics of Computation*, 80(275):1287–1301, 2011.

[58] Qun Lin and Hehu Xie. A multi-level correction scheme for eigenvalue problems. *Mathematics of Computation*, 84(291):71–88, 2015.

[59] Hao Li and Yidu Yang. The adaptive finite element method based on multi-scale discretizations for eigenvalue problems. *Computers & Mathematics with Applications*, 65(7):1086–1102, 2013.

[60] Yidu Yang and Hai Bi. Two-grid finite element discretization schemes based on shifted-inverse power method for elliptic eigenvalue problems. *SIAM Journal on Numerical Analysis*, 49(4):1602–1624, 2011.

[61] Roland Pabel. *Adaptive Wavelet Methods for Variational Formulations of Nonlinear Elliptic PDEs on Tensor-Product Domains*. Logos Verlag Berlin GmbH, 2015.

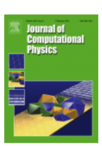
[62] Jochen Garcke. Sparse grids in a nutshell. In *Sparse grids and applications*, pages 57–80. Springer, 2012.

[63] Nabi Chegini, Stephan Dahlke, Ulrich Friedrich, and Rob Stevenson. Piecewise tensor product wavelet bases by extensions and approximation rates. *Mathematics of Computation*, 82(284):2157–2190, 2013.

Appendix A

Copyright Documentation

A.1 Copyright Documentation of Chapter 2



Recursive integral method for transmission eigenvalues

Author: Ruihao Huang,Allan A. Struthers,Jiguang Sun,Ruming Zhang

Publication: Journal of Computational Physics

Publisher: Elsevier

Date: 15 December 2016

Please note that, as the author of this Elsevier article, you retain the right to include it in a thesis or dissertation, provided it is not published commercially. Permission is not required, but please ensure that you reference the journal as the original source. For more information on this and on your other retained rights, please visit: https://www.elsevier.com/about/our-business/policies/copyright#Author-rights

A.2 Copyright Documentation of Chapter 3

License Number	4721511001510
License date	Dec 03, 2019
Licensed Content Publisher	John Wiley and Sons
Licensed Content Publication	Numerical Linear Algebra With Applications
Licensed Content Title	Recursive integral method with Cayley transformation
Licensed Content Author	C. Yang, J. Sun, R. Huang
Licensed Content Date	Jul 10, 2018
Licensed Content Volume	25
Licensed Content Issue	6
Licensed Content Pages	12
Type of use	Dissertation/Thesis
Requestor type	Author of this Wiley article
Format	Electronic

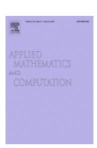
12/3/2019 RightsLink Printable License

Portion	Full article
Will you be translating?	No
Title of your thesis / dissertation	Novel Computational Methods For Eigenvalue Problems
Expected completion date	Dec 2019
Expected size (number of pages)	100
Requestor Location	Ruihao Huang 1807 Woodmar Dr, Apt D HOUGHTON, MI 49931 United States Attn: Ruihao Huang
Publisher Tax ID	EU826007151
Total	0.00 USD
Terms and Conditions	

A.3 Copyright Documentation of Chapter 4

A Memory Efficient Multilevel Spectral Indicator Method has not yet been published (up to date),there is no need to obtain copyright documentation.

A.4 Copyright Documentation of Chapter 5



A new fast method of solving the high dimensional elliptic eigenvalue problem

Author: Ruihao Huang,Lin Mu
Publication: Applied Mathematics and Computation
Publisher: Elsevier
Date: 1 September 2019

Please note that, as the author of this Elsevier article, you retain the right to include it in a thesis or dissertation, provided it is not published commercially. Permission is not required, but please ensure that you reference the journal as the original source. For more information on this and on your other retained rights, please visit: https://www.elsevier.com/about/our-business/policies/copyright#Author-rights