The Spectral Transformation Lanczos Algorithm for the Symmetric-Definite Generalized Eigenvalue Problem: A Comparative Analysis with Conditioning Insights

by

Ayobami Adebesin

Under the Direction of Michael Stewart, Ph.D.

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of

Master of Science

in the College of Arts and Sciences

Georgia State University

2025

ABSTRACT

This thesis investigates the application of the spectral transformation Lanczos algorithm (ST-Lanczos) to a generalized symmetric-definite eigenvalue problem involving real symmetric matrices A and B, with B being positive definite and possibly ill conditioned. The Lanczos algorithm is a well-known iterative algorithm for computing the eigenvalues of a symmetric matrix and it works well for finding the extreme points in the spectrum. By leveraging a shifted and inverted formulation of the problem, the ST-Lanczos algorithm relies on iterative projection to approximate extremal eigenvalues near a shift σ. While previous work has been done using direct methods, the goal of this thesis is to use an iterative approach, and analyze how the error bounds already proven for direct methods play out in an iterative context.

This study focuses primarily on benchmarking the ST-Lanczos method against established direct methods in the literature and addresses challenges in numerical stability, computational efficiency, and sensitivity of residuals to ill-conditioning.

INDEX WORDS: eigenvalues, eigenvectors, Lanczos algorithm, Ritz values, Krylov subspaces, spectral transformation, orthogonality

The Spectral Transformation Lanczos Algorithm for the Symmetric-Definite Generalized Eigenvalue Problems: A Comparative Analysis with Conditioning Insights

by

Ayobami Adebesin

Committee Chair:	Michael Stewart
Committee:	Russell Jeter
	Vladimir Bondarenko

Electronic Version Approved:

Office of Graduate Studies

College of Arts and Sciences

Georgia State University

May 2025

DEDICATION

To my loving parents, Mr. and Mrs. Adebesin —This work is a tribute to the sacrifices you made, the lessons you taught me, and for giving up your dreams so that I could chase mine. For offering me your best, your love and the boundless faith you have in me.

ACKNOWLEDGMENTS

First and foremost, I express my profound gratitude to God for the gift of life, grace and opportunity bestowed onto me for bringing me this far in life and helping me complete another step towards achieving my dreams.

I would also like to express my deepest gratitude to my thesis advisor, Professor Michael Stewart for his unwavering support, guidance and impact on the completion of this thesis. I had absolutely zero knowledge or idea on this subject before starting this thesis, but his expertise, patience and insightful feedback have been invaluable in shaping this thesis and my growth as a mathematician. It is such a great privilege to have had the opportunity to learn from you. God bless you sir.

Additionally, I would also like to extend my sincere appreciation to the members of my committee, Professor Russell Jeter, and Professor Vladimir Bodarenko, for their time, thoughtful suggestions and feedback on this work. Your work have greatly influenced my research and your perspectives have enriched this thesis and helped me refine my ideas.

To my colleagues and friends in the department — John Ajayi, Xavier Sodjavi, Sheriff Akeeb, Akinwale Famotire, Emeka Mazi, to mention a few, I say a big thank you for creating a highly simulating and collaborative environment for learning. My sincere gratitude also goes to the faculty members of the Department of Mathematics at Georgia State University, many of whom I have had the honor of learning from. I am particularly thankful to Dr. Zhongshan Li, Professor Alexandra Smirnova, and Professor Mariana Montiel, to mention a few, for

their mentorship, expertise, and encouragement throughout my academic journey. Their dedication to teaching and research has been a constant source of inspiration. Additionally, I extend my thanks to the entire staff of the department for their support and assistance, which have been instrumental in creating a conducive environment for learning and research.

Finally, I would like to acknowledge my parents, Mr. and Mrs. Adebesin, my siblings and several father and mother figures in my life — Mr. and Mrs. Olawuyi, Mrs. Abioye, Mr. Agboola for their unwavering support. God bless you all.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 Background

The problem of computing eigenvalues and eigenvectors of matrices in numerical linear algebra is a well-studied one. The computation of eigenvalues and eigenvectors plays a central role in scientific computing with applications in structural analysis, quantum mechanics, data science and control theory. However, eigenvalue problems (standard and generalized) involving dense and sparse matrices present significant computational challenges, especially as the size of the matrices increases. These problems are fundamental in many scientific and engineering disciplines where the underlying mathematical models are often expressed in terms of eigenvalue equations. Historically, methods for solving eigenvalue problems date back to the early 20th century with foundational contributions from David Hilbert, Erhard Schmidt, and John von Neumann, who laid the groundwork for understanding linear operators and their spectral properties.

With the advent of digital computing in the mid-20th century, numerical methods for eigenvalue problems began to flourish. Classical iterative methods, such as the power iteration and inverse iteration, were among the first to be employed due to their simplicity and effectiveness for small-scale problems. However, as computational requirements grew, particularly with the need to solve larger sparse systems, researchers turned to more sophisticated algorithms. The Lanczos method, introduced by Cornelius Lanczos in 1950, represented a significant advancement for efficiently solving eigenvalue problems for large symmetric ma-

trices. The method exploits the sparsity of matrices and reduces the dimensionality of the problem by constructing a tridiagonal matrix whose eigenvalues approximate those of the original matrix.

An important class of eigenvalue problems, which is the main focus of this thesis, is the generalized eigenvalue problem (GEP). The GEP takes the form $A\mathbf{v} = \lambda B\mathbf{v}$ where A and B are square matrices, λ is a generalized eigenvalue, and $\mathbf{v} \neq \mathbf{0}$ is the corresponding generalized eigenvector. This class of problems arises naturally in a number of application areas, including structural dynamics, data analysis and has a long history in the research literature on numerical linear algebra.

1.2 Literature Review

Generalized eigenvalue problems involving symmetric and positive definite matrices are fundamental in numerical linear algebra with applications in structural dynamics, quantum mechanics, and control theory. Solving these kind of problems involve computing the eigenvalues λ and eigenvectors $\mathbf{v}$ that satisfies the equation. The choice of method depends on the properties of the matrix involved in the problem we are trying to solve (e.g, sparsity, symmetry) and computational constraints. In this section, we discuss some of the research that has been done on this topic.

When B is invertible, the problem is reduced to $B^{-1}A\mathbf{v} = \lambda\mathbf{v}$. However, explicity forming $B^{-1}A$ is numerically unstable if B is ill-conditioned. Since B a symmetric and positive definite B, one can compute a Cholesky factorization $B = LL^T$ which allows us to

reduce the equation to a standard eigenvalue problem $L^{-1}AL^{-T}\mathbf{y} = \lambda\mathbf{y}$ where $\mathbf{y} = L^T\mathbf{v}$, which can then be solved by using the symmetric QR algorithm to compute a diagonal eigenvalue~~Schur~~ decomposition. A detailed treatment for this case is presented in [2]. In practice, this approach usually results in small relative residuals for eigenvalues that are large in magnitude and larger relative residuals for eigenvalues that are~~eigenvalue~~ smaller in magnitude.

The QZ algorithm [3] for the non-symmetric GEP, is an iterative method that generalizes the QR algorithm to handle directly the generalized eigenvalue problem instead of the standard eigenvalue problem, thereby avoiding potential problems with inverting ill-conditioned or singular B.~~singular B~~ It applies orthogonal transformations to simutaneously reduce A and B to upper triangular forms from which the eigenvalues are extracted. Although this method is robust and backward stable, it is computationally expensive, thereby limiting its use to small or medium sized matrices.

A more stable approach for a dense symmetric semidefinite problem where B is symmetric positive definite and possibly ill-conditioned is described in [4]. This approach uses a spectral transformation method that leverages a shifted and inverted formulation of the problem, which it then solves using symmetric factorizations. Using a rook pivoted LDL^T factorization for $(A - \sigma B)$ and a Cholesky factorization for B, such that $(A - \sigma B) = C_a D_a C_a^T$ and $B = C_b C_b^T$, the generalized problem is transformed to a standard one given by

$$C_b^T C_a^{-T} D_a C_a^{-1} C_b \mathbf{u} = \theta \mathbf{u}. \tag{1.1}$$

Under some technical assumptions that typically hold in practice, it was proved that if

the scaled shift

$$\sigma_0 = \sigma \frac{\|B\|}{\|A\|} \tag{1.2}$$

is not too large and the shift σ is not chosen to be too close to a generalized eigenvalue, then this approach is stable in the sense that it gives generalized eigenvalues of a pair of matrices close to A and B. This is true regardless of any ill conditioning in $A - \sigma B$. This approach was validated by numerical experiments, contrasting it with Cholesky-based methods that are unstable for small eigenvalues.

1.3 Mathematical Preliminaries

In this section, we shall introduce some notations and the key mathematical concepts underlying the eigenvalue problems that will be used throughout this study.

1.3.1 Notation

Throughout this study, we make use of the following notations:

$A \in \mathbb{R}^{m \times n}$: denotes a matrix

$[A]_{ij}$: denotes element (i, j) of A

$\mathbf{x} \in \mathbb{R}^m$: denotes a column vector

A^T : denotes the transpose of matrix A

$\| \cdot \|$: denotes a vector or matrix norm

$A_{i:i',j:j'}$: denotes the $(i' - i + 1) \times (j' - j + 1)$ submatrix of A

1.3.2 Floating Point Arithmetic

We define a *floating point* number system, $\mathbf{F}$ as a bounded subset of the real numbers $\mathbb{R}$, such that the elements of $\mathbf{F}$ are the number 0 together with all numbers of the form

$$x = \pm(m/\beta^t)\beta^e,$$

where m is an integer in the range $1 \leq m \leq \beta^t$ known as the significand, $\beta \geq 2$ is known as the *base* or *radix* (typically 2), e is an arbitrary integer known as the exponent and $t \geq 1$ is known as the precision.

To ensure that a nonzero element $x \in \mathrm{F}$ is unique, we can restrict the range of $\mathbf{F}$ to $\beta^{t-1} \leq m \leq \beta^t - 1$. The quantity $\pm(m/\beta^t)$ is then known as the *fraction* or *mantissa* of x. We define the number $u := \frac{1}{2}\beta^{1-t}$ as the *unit roundoff* or *machine epsilon*. In a relative sense, the *unit roundoff* is as large as the gaps between floating point numbers get.

Let $fl : \mathbb{R} \to \mathbf{F}$ be a function that gives the closest floating point approximation to a real number, then the following theorem gives a property of the unit roundoff.

Theorem 1.3.1. *If $x \in \mathbb{R}$ is in the range of $\mathbf{F}$, then $\exists\ \epsilon$ with $|\epsilon| \leq u$ such that $fl(x) = x(1+\epsilon)$.*

One way we could think of this is that, the difference between a real number and its closest floating point approximation is always smaller than u in relative terms.

1.3.3 Conditioning and Stability

Given any mathematical problem $f : X \to Y$, the conditioning of that problem pertains to inherent sensitvity of the problem, while stability of the algorithm pertains to the propagation of errors in an algorithm used in solving that problem on a computer. A *well-conditioned* problem is one with the property that small perturbations of the input lead to only small changes in the output. An *ill-conditioned* problem is one with the property that small perturbations in the input leads to a large change in the output.

For any mathematical problem, we can associate a number called the *condition number* to that problem that tells us how well-conditioned or ill-conditioned the problem is. For the purpose of this thesis, we shall only be considering the condition number of matrices. Since matrices can be viewed as linear transformations from one vector space to another, it makes sense to define a condition number for matrices.

For a matrix $A \in \mathbb{R}^{m \times n}$, the condition number with respect to a given norm is defined as

$$\kappa(A) = \|A\| \cdot \|A\|^{-1}. \tag{1.3}$$

In simpler terms, the condition number quantifies how the relative error in the solution of a linear system $A\mathbf{x} = \mathbf{b}$ can be amplified when there is a small perturbation in the input vector $\mathbf{x}$. If $\kappa(A)$ is small, A is said to be *well-conditioned*; if $\kappa(A)$ is large, then A is said to be *ill-conditioned*. It should be noted that the notion of being "small" or "large" depends on the application or problem we are solving. If $\|\cdot\| = \|\cdot\|_2$ (spectral norm or 2-norm), then

$\|A\| = \sigma_1$ and $\|A^{-1}\| = 1/\sigma_m$, so that

$$\kappa(A) = \frac{\sigma_1}{\sigma_m}, \tag{1.4}$$

where σ_1 and σ_m are the largest and smallest singular values of A respectively. Throughout the remainder of this thesis, unless stated otherwise, $\|\cdot\|$ will refer to the spectral norm, or 2-norm.

1.3.4 The Generalized Eigenvalue Problem

Let $A, B \in \mathbb{R}^{m\times m}$, be any general square matrices. A *pencil* is an expression of the form $A - \lambda B$, with $\lambda \in \mathbb{C}$. The *generalized eigenvalues* of $A - \lambda B$ are the elements of the set $\Lambda(A, B)$ defined by

$$\Lambda(A, B) = \{z \in \mathbb{C} : \det(A - zB) = 0\}. \tag{1.5}$$

In other words, the generalized eigenvalues of A and B are the roots of the characteristic polynomial of the pencil $A - \lambda B$ given by

$$p_{A,B}(\lambda) = \det(A - \lambda B) = 0. \tag{1.6}$$

A pencil is said to be *regular* if there exists at least one value of $\lambda \in \mathbb{R}$ such that $\det(A-\lambda B) = 0$, otherwise it is called *singular*.

If $\lambda \in \Lambda(A, B)$ and $0 \neq \mathbf{v} \in \mathbb{C}^m$ satisfies

$$A\mathbf{v} = \lambda B\mathbf{v}, \tag{1.7}$$

then $\mathbf{v}$ is a generalized eigenvector of A and B corresponding to λ. The problem of finding

non-trivial solutions to (1.7) is known as the *generalized eigenvalue problem.*

If B is non-singular, then the problem reduces to a standard eigenvalue problem

$$B^{-1}A\mathbf{v} = \lambda\mathbf{v}. \tag{1.8}$$

In this case, the generalized eigenvalue problem has m eigenvalues if $\text{rank}(B) = m$. This suggests that the generalized eigenvalues of A and B are equal to the eigenvalues of $B^{-1}A$. If B is singular or rank deficient, then the set of generalized eigenvalues $\Lambda(A, B)$ may be finite, empty or infinite. If $\Lambda(A, B)$ is finite, the number of eigenvalues will be less than m. This is because the characteristic polynomial $\det(A - \lambda B)$ is of degree less than m, so that there is not a complete set of eigenvalues for the problem.

If A and B have a nontrivial common null space, then every choice of λ will be a solution to (1.7). In this case, we say that the pencil $A - \lambda I$ is *singular.* Otherwise, we say that the pencil is *regular.* For the purpose of this study, we shall assume that A and B do not have a nontrivial common null space, that is

$$\mathcal{N}(A) \cap \mathcal{N}(B) = \{\mathbf{0}\}. \tag{1.9}$$

When A and B are symmetric and B is positive definite, we shall call the problem symmetric-definite generalized eigenvalue problem, which will be the focus of this thesis. The symmetric definite generalized eigenvalue problem is fully analogous to the symmetric standard eigenvalue problem. In this case we will see that the pencil is regular, the generalized eigenvalues are all real, and the generalized eigenvectors are orthogonal with respect to the inner product $\langle \mathbf{x}, \mathbf{y} \rangle_B = \mathbf{y}^T B\mathbf{x}$.

1.3.5 Generalized Eigenvalue Problem with Symmetry

In this section, we shall consider the symmetric version of the generalized eigenvalue problem that was described in Section 1.3.4, which will be the main focus of this thesis. We will also consider the methodological approach used in solving the problem, and discuss the challenges involved in solving these kind of problems.

The symmetric-definite generalized eigenvalue problem is formally given by:

$$A\mathbf{v} = \lambda B\mathbf{v}, \qquad \mathbf{v} \neq 0 \tag{1.10}$$

where $A \in \mathbb{R}^{m\times m}$ is symmetric and $B \in \mathbb{R}^{m\times m}$ is symmetric positive definite.

Problem (1.10) can be reformulated as

$$\beta A\mathbf{v} = \alpha B\mathbf{v}, \qquad \mathbf{v} \neq 0 \tag{1.11}$$

We have replaced λ with α/β for convenience so that the generalized eigenvalues will be of the form (α, β). If $\beta = 0$, then the generalized eigenvalues $\Lambda(A, B)$ will be infinite. The formulation using equation(1.11) is useful when describing the error bounds, as we shall later see. We shall alternate between (1.10) and (1.11) when convenient.

It is known that the eigenvalues of a symmetric definite generalized eigenvalue problem are real. An interesting property of the symmetric-definite problem is that it can be transformed to an equivalent symmetric definite problem with a congruence transformation. Let P be a non-singular matrix, then (1.10), is equivalent to the transformed equation

$$(P^T AP)\mathbf{v} = \lambda(P^T BP)\mathbf{v}, \tag{1.12}$$

so that

$$\Lambda(A, B) = \Lambda(P^T AP, P^T BP). \tag{1.13}$$

Furthermore, for any symmetric-definite pair (A, B), A and B can be simultaneously diagonalized by a non-singular matrix P such that

$$P^T AP = D_a \qquad \text{and} \qquad P^T BP = D_b, \tag{1.14}$$

where $D_a = \text{diag}(a_1, \ldots a_n)$, and $D_b = \text{diag}(b_1, \ldots b_n)$. The generalized eigenvalues λ of A and B will be the diagonal elements of $D_b^{-1} D_a$, or put simply $\lambda_i = a_i/b_i$, and the eigenvectors will be the columns of P. The existence of P when B is semidefinite is given in [2, p. 498].

To compute the set of generalized eigenvalues $\Lambda(A, B)$ that satisfies (1.10), our approach in this thesis, similar to what was used in [4] and [1], is to transform the problem into a standard eigenvalue problem using a spectral transformation, after which we apply an iterative algorithm, like the Lanczos algorithm to compute the eigenvalues. In practice, we often compute a subset of these generalized eigenvalues corresponding to those in the vicinity of a given shift σ. To have a deep understanding of this approach, the next section discusses this approach in detail, focusing on the relationship between the eigenvalues of the original problem and the eigenvalues of the transformed problem.

1.3.6 Spectral Transformation

Spectral transformation in numerical linear algebra is a technique that is used to modify the spectrum of matrix in a controlled way. This is usually done to improve the convergence properties of an algorithm or to make certain matrix properties more accessible. In the

context of eigenvalue problems, spectral transformation is often used in direct and iterative methods, where manipulating the matrix can help focus on certain eigenvalues or improve numerical stability.

The central idea behind spectral transformation is that by applying a rational or polynomial transformation to the matrix A, we can manipulate its eigenvalues to increase the magnitude of the eigenvalues we are interested in without changing their eigenvectors. There are various types of spectral transformation, but the one of particular interest in this thesis is the *shift-invert* transformation. The shift-invert transformation involves transforming the original problem into a shifted and inverted one which can then be solved using a direct or iterative solver. This method focuses on finding the eigenvalues near a specified shift σ. It is useful when one is interested in a few eigenvalues near a given point in the spectrum.

Consider the problem of computing the eigenvalues of a matrix $A \in \mathbb{R}^{m\times m}$. Assume that m is so large that computing all the eigenvalues of A is not computationally feasible but rather, we are interested in computing the eigenvalues in a certain region of the spectrum of A. We can pick a shift $\sigma \in \mathbb{R}$ that is not an eigenvalue of A. The shifted and inverted matrix is then given by $(A - \sigma I)^{-1}$. The eigenvectors of $(A - \sigma I)^{-1}$ are the same as the eigenvectors of A, and the corresponding eigenvalues are $(\lambda_j - \sigma)^{-1}$, for each eigenvalue λ_j of A. This shifts the spectrum of A, making the eigenvalues near σ much more prominent in the transformed matrix. This shifting strategy can be used in other iterative algorithms like the inverse iteration [5].

For the generalized eigenvalue problem given in (1.10), if we introduce a shift $\sigma \in \mathbb{R}$ so

that $A - \sigma B$ is non singular, a simple shifted and inverted formulation of the problem is given by

$$(A - \sigma B)^{-1} B\mathbf{v} = \theta \mathbf{v}, \tag{1.15}$$

where $\theta = 1/(\lambda - \sigma)$. Note, however, that this formulation does not result in a symmetric standard eigenvalue problem.

Suppose σ is close enough to a generalized eigenvalue $\lambda_J \in \Lambda(A, B)$ much more than the other generalized eigenvalues, then $(\lambda_J - \sigma)^{-1}$ may be much larger than $(\lambda_j - \sigma)^{-1}$ for all $j \neq J$. This transformation will map the eigenvalues in the neighborhood of σ to the extreme part of the new spectrum, which can be favorable for the convergence of Krylov methods. However there is a problem. The formulation (1.15) is not symmetric, so we cannot use the Lanczos algorithm. We now consider a shift-and-invert formulation that preserves symmetry. Since B is positive definite, we can compute a Cholesky factorization $B = C_b C_b^T$. Given this decomposition, we can have a formulation that preserves symmetry, guaranteed by the following lemma

Lemma 1.3.2. *Let $A - \sigma B$ be nonsingular and $B = C_b C_b^T$, where C_b is a lower triangular matrix. Assume that $\lambda \neq \infty$ and $\mathbf{v} \neq \mathbf{0}$ satisfies (1.10). Then $\theta = 1/(\lambda - \sigma)$ is an eigenvalue of the problem*

$$C_b^T (A - \sigma B)^{-1} C_b \mathbf{u} = \theta \mathbf{u}, \qquad \mathbf{u} \neq \mathbf{0}, \tag{1.16}$$

with eigenvector $\mathbf{u} = C_b^T \mathbf{v} \neq \mathbf{0}$.

Conversely, assume that $\mathbf{u} \neq \mathbf{0}$ is an eigenvector for (1.16), with eigenvalue θ. If $C_b \mathbf{u} \neq$

$\mathbf{0}$, *then the eigenvector* $\mathbf{v} = (A - \sigma B)^{-1} C_b \mathbf{u} \neq \mathbf{0}$ *is an eigenvector for (1.11) with eigenvalue* $(1 + \sigma\theta, \theta)$. *In this case, with* $\mathbf{v}$ *defined in ths way, we have* $C_b^T \mathbf{v} = \theta \mathbf{u}$. *If instead we have*$C_b \mathbf{u} = \mathbf{0}$, *then* $\theta = 0$ *and* $(1, 0)$ *is an eigenvalue for (1.11) with eigenvector given by* $\mathbf{v} = \mathbf{u}$. *If* C_b *is* $m \times m$ *and invertible, then we have* $C_b \mathbf{u} \neq \mathbf{0}$ *and can use the alternative formula* $\mathbf{v} = C_b^{-T} \mathbf{u}$ *to obtain an eigenvector of (1.11).*

The proof of this lemma can be found in [4], although most claims of the lemma were used in [1]. In essence, Lemma 1.3.2 describes the relationship between the eigenvalues (*resp.* eigenvectors) of the original problem (1.10) and the eigenvalues (*resp.* eigenvectors) of the spectral transformed problem (1.16). Equation (1.16) gives us the spectral transformed version of the original generalized problem that preserves symmetry. Since the problem is now in a standard form, we can then apply the Lanczos algorithm to compute the desired eigenvalues within the neighborhood of σ, together with their corresponding eigenvectors. It should be noted that forming the spectral matrix in (1.16) is not desirable in a realistic problem since it does not preserve sparsity and will be very inefficient on most realistic problems. In reality, we employ stable factorizations of $(A - \sigma B)^{-1}$ that preserves symmetry in order to solve (1.16) effectively. Our choice of factorizations will be discussed in later chapters. For now, we turn our attention to the Krylov subspace method that will be employed in solving (1.16).

1.3.7 Lanczos Algorithm

The Lanczos algorithm is an iterative method in numerical linear algebra used in finding the eigenvalues and eigenvectors of a *symmetric* matrix. It is particularly useful when dealing

with large scale problems, where directly computing the eigenvalues and eigenvectors of the matrix would be computationally expensive of infeasible. It works by finding the "most useful" eigenvalues of the matrix — typically those at the extreme of the spectrum, and their eigenvectors. At it's core, the main goal of the algorithm is to approximate the extreme eigenvalues and eigenvectors of a large, sparse, symmetric matrix by transforming the matrix into a smaller tridiagonal matrix that preserves the extremal spectral properties of the original matrix. This reduction is achieved by iteratively constructing an orthonormal basis of the Krylov subspace associated with the matrix.

Given a symmetric matrix $A \in \mathbb{R}^{m\times m}$, and an initial vector $\mathbf{v}_1$, the Lanczos algorithm produces a sequence of vectors $\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_n$ (where n is the number of iterations) that forms an orthonormal basis for the n-dimensional Krylov subspace

$$\mathcal{K}_n(A, \mathbf{v}_1) = \text{span}(\{\mathbf{v}_1, A\mathbf{v}_1, A^2\mathbf{v}_1, \ldots, A^{n-1}\mathbf{v}_1\}). \tag{1.17}$$

This orthonormal basis is used to form a tridiagonal matrix T_n whose eigenvalues approximate the eigenvalues of A.

1.4 Motivation of Study

This study is motivated by several key factors that underscore the importance of advancing our understanding and capabilities in solving these type of problems. Originally, the motivation for this study arises from the need to compare the efficiency, accuracy and stability of iterative and direct methods for solving eigenvalue problems. In particular, the proven error bounds for the direct method in the paper by [4], shows that for a shift of moderate size,

the relative residuals are small for generalized eigenvalues that are not much larger than the shift. It is natural to ask if the same can be said for an iterative method like the lanczos algorithm.

On another hand, the motivation is based on the goal of advancing the field of numerical linear algebra. The insights gained from analyzing the ST-Lanczos algorithm for generalized eigenvalue problems may inform the development of new algorithms or hybrid methods that combine the strengths of different methods. This could potentially lead to breakthroughs in the development of eigenvalue algorithms that are more reliable than the current ones we have today.

CHAPTER 2
METHODOLOGY AND ALGORITHM DESCRIPTION

In this chapter, we shall present a detailed description of the methodologies and implementation of algorithms used in this thesis to solve the generalized eigenvalue problem. We begin by describing the problem setup, followed by a discussion of the algorithms used, together with their implementation details. This chapter aims to provide a comprehensive understanding of how these algorithms are applied to derive the solutions to the problem at hand. We shall also give a description of the numerical experiments we setup to investigate the efficiency of these algorithms.

To compute the eigenvalues and eigenvectors that satisfy (1.10) with spectral transformation lanczos algorithm, our approach will be in two steps:

- Transform the generalized problem into a spectral transformed standard eigenvalue problem.
- Solve the spectral problem with Lanczos algorithm.

As described in Section 1.3.6, Lemma 1.3.2 gives us the relationship between the eigenvalues (*resp.* eigenvectors) of the original problem and the eigenvalues (*resp.* eigenvectors) of the spectral transformed problem. We also recall that solving 1.16 relies on symmetric factorization of the $A - \sigma B$ and B. The following section describes the possible ways of doing this.

2.1 Stable Decompositions

We shall begin by computing decompositions for $A - \sigma B$ and B. If B is positive definite, we can compute a Cholesky decomposition $B = C_b C_b^T$ using the SciPy `cholesky` method which calls LAPACK `xPOTRF`. However, if B is semi positive definite, this function call fails and we use the more robust pivoted Cholesky factorization `xPSTRF` by calling the inbuilt LAPACK bindings in SciPy.

There are various possible factorization options for $A - \sigma B$. One option is to use the pivoted LDL^T factorization used by [4] and [1] where D is a block diagonal matrix with 1×1 and 2×2 on the diagonal, and L is a lower triangular matrix. This factorization ideally uses the "rook pivoting" scheme, which is stable. Although the standard LDL^T factorization (without "rook pivoting") is available in the SciPy linear algebra module, there is no option to use the rook pivoting scheme except if one chooses to write a custom LAPACK binding that makes use of `DSYTRF_ROOK`. While this can guarantee some stability for the problem we are trying to solve, it usually involves extra work in processing the 2×2 blocks to make D diagonal.

Another possible choice of factorization is an eigenvalue decomposition of $A - \sigma B$. If we use a symmetric eigenvalue decomposition $A - \sigma B = WDW^T$, our numerical experiments reveals that this stabilizes the Ritz residuals and generalized form of the residuals together with the advantage that these residuals are insensitive to ill-conditioning in $A - \sigma B$. This can be done using inbuilt eigenvalue solvers in SciPy or any linear algebra library. This is the most promising factorization, however computing eigenvalue decompositions for large

sparse problems become computationally expensive and not feasible in reality. So, although the eigenvalue decomposition allows tests of stability, in a real implementation it would be replaced by an LDL^T factorization.

Lastly, we could make use of an LU factorization for $A - \sigma B$. Unlike the previous factorizations, the stability for the Ritz residuals is not as great, as we observe that they depend on the conditioning of $A - \sigma B$. However, for the purpose of this thesis, we make use of the LU decomposition since it is computationally less expensive and easy to use and implement. Comparing it to the eigenvalue decomposition of $A - \sigma B$ also illustrates a potential mechanism of instability in implementing the spectral shift.

One major takeaway from our experiments with the various options of factorizing $A - \sigma B$ is that symmetry is clearly important for stability. We plan to give a mathematical justification for this in future work.

Given our recent results, one might suggest using a stable decomposition such as the LU factorization. However, as we will demonstrate, decompositions that preserve symmetry exhibit certain stability advantages in practice.

2.2 The Lanczos decomposition

In this section, we revisit the Lanczos algorithm, and discuss how we apply it to the spectral transformed problem. As discussed in section (1.3.7), the Lanczos algorithm approximates the eigenvalues of the original problem by projecting it onto a Krylov subspace spanned by successive powers of the system matrix applied to an initial vector. The eigenvalue

approximations arises from the tridiagonal matrix obtained through the Lanczos process, which captures the essential spectral characteristics of the original matrix.

Given $A \in \mathbb{R}^{m\times m}$, with $A = A^T$, the pesudocode for the lanczos algorithm is described by Algorithm 1. After the completion of Algorithm 1, the γ_j and δ_j are used to construct the

Algorithm 1 Lanczos Algorithm for a Symmetric Matrix

Require: $A = A^T$, number of iterations: n, tolerance: tol

function LANCZOS(A, n, tol)
  Choose an arbitrary vector $\mathbf{b}$ and set an initial vector $\mathbf{q_1} = \mathbf{b}/\|\mathbf{b}\|_2$
  Set $\delta_0 = 0$ and $\mathbf{q_0} = \mathbf{0}$
  for $j = 1, 2, \ldots, n$ **do**
    $\mathbf{v} = A\mathbf{q_j}$
    $\gamma_j = \mathbf{q_j}^T\mathbf{v}$
    $\mathbf{v} = \mathbf{v} - \delta_{j-1}\mathbf{q_{j-1}} - \gamma_j\mathbf{q_j}$
    Full reorthogonalization: $\mathbf{v} = \mathbf{v} - \sum_{i\leq j}(\mathbf{q_i}^T\mathbf{v})\mathbf{q_i}$
    $\delta_j = \|\mathbf{v}\|_2$
    if $\delta_j < tol$ **then**
      restart or **exit**
    end if
    $\mathbf{q_{j+1}} := \mathbf{v}/\delta_j$
  end for
end function

tridiagonal matrix $T_n \in \mathbb{R}^{n\times n}$ and the vectors $\mathbf{q}_j$ are stacked together to form an orthogonal matrix $Q_n \in \mathbb{R}^{m\times n}$ given by:

$$T_n = \begin{pmatrix} \gamma_1 & \delta_1 & & & \\ \delta_1 & \gamma_2 & \delta_2 & & \\ & \delta_2 & \gamma_3 & \delta_3 & \\ & & \ddots & \ddots & \vdots \\ & & & \delta_{n-1} & \gamma_n \end{pmatrix}$$

$$Q_n = \left[\begin{array}{c|c|c|c} & & & \\ \mathbf{q_1} & \mathbf{q_2} & \cdots & \mathbf{q_n} \\ & & & \end{array}\right].$$

The decomposition is given by

$$AQ_n = Q_n T_n + \delta_n \mathbf{q_{n+1}} \mathbf{e_n}^T \tag{2.1}$$

In theory, the vectors q_j's should be orthonormal, but due to floating-point errors, there will be loss of orthogonalization, hence the need for line 8 in Algorithm 1.

Let $\theta_i, i = 1, 2, \ldots n$ (which can be computed by standard functions using any eigenvalue solver) be the eigenvalues of T_n, and $\{\mathbf{y_i}\}_{i=1:n}$ be the associated eigenvectors. The $\{\theta_i\}$ are called the *Ritz values* and the vectors $\{Q_n \mathbf{y_i}\}_{i=1:n}$ are called the *Ritz vectors*. Hence, the eigenvalues of A are on both ends of the are well approximated by the Ritz values, with the Ritz vectors as their approximate corresponding eigenvectors of A.

Since the generalized eigenvalue problem we started with has been reduced to a standard one as shown in equation (1.16), Algorithm 1 can be applied to equation (1.16) with some slight modifications. The spectral form of Algorithm 1 is given by Algorithm 2. After applying the lanczos procedure to the spectral transformed problem (1.16), we compute the converged Ritz pairs using a certain tolerance. The converged Ritz pairs are then mapped to the generalized eigenvalues and eigenvectors where we can observe the behaviour of these residuals with respect to conditioning.

2.3 Problem Setup

To evaluate the performance and robustness of the spectral transformation lanczos algorithm, we setup a problem with predetermined eigenvalues, use the algorithm to compute the eigenvalues, and show that the residuals follow closely with the bounds predicted for

Algorithm 2 Spectral Lanczos Algorithm for (1.16)

Require: $A = A^T$, $B = B^T$, with B being positive definite or semidefinite
Require: number of iterations: n, size of matrix A or B: m, tolerance: tol
Require: $\sigma \in \mathbb{R}$: shift not close to a generalized eigenvalue

function SPECTRAL_LANCZOS(A, B, m, n, σ, tol)
    Choose an arbitrary vector $\mathbf{b}$ and set an initial vector $\mathbf{q_1} = \mathbf{b}/\|\mathbf{b}\|_2$
    Set $\beta_0 = 0$ and $\mathbf{q_0} = \mathbf{0}$
    Set $Q = \text{zeros}(m, n+1)$
    Precompute the LU factorization of $A - \sigma B$: $LU = (A - \sigma B)$
    Factor: $B = CC^T$
    for $j = 1, 2, \ldots, n$ **do**
        $Q[:, j] = \mathbf{q}_j$
        $\mathbf{u} = C\mathbf{q}_j$
        Solve: $(LU)\mathbf{v} = \mathbf{u}$ for $\mathbf{v}$
        $\mathbf{v} = C^T\mathbf{v}$
        if $j < n$ **then**
            $\alpha_j = \mathbf{q}_j^T\mathbf{v}$
            $\mathbf{v} = \mathbf{v} - \beta_{j-1}\mathbf{q}_{j-1} - \alpha_j\mathbf{q}_j$
            Full reorthogonalization: $\mathbf{v} = \mathbf{v} - \sum_{i \le j}(\mathbf{q}_i^T\mathbf{v})\mathbf{q}_i$
            $\beta_j = \|\mathbf{v}\|_2$
            if $\beta_j < tol$ **then**
                restart or **exit**
            end if
            $\mathbf{q}_{j+1} := \mathbf{v}/\beta_j$
        end if
    end for
    $Q = Q[:, : n]$
    $\mathbf{q} = Q[:, n]$
    return $(Q, T, \mathbf{q})$
end function

direct methods in [4]. While there are other options of using matrices from open source repositories like Matrix Market, we choose to use this approach so that we can control the size, condition number and other properties of the matrix so as to observe the effect of this properties on the algorithm.

Starting with a diagonal matrix $D \in \mathbb{R}^{m \times m}$ with known eigenvalues, we generate a

random matrix P of size $m \times m$ with standard normal distribution. Since the QR factorization is guaranteed to exist for any matrix, we take the QR factorization of P to obtain an orthogonal matrix Q, which is used to create a matrix C using orthogonal transformation. Hence $C = QDQ^T$ is unitarily similar to D.

Next, we initialize a random lower triangular matrix $L_0 \in \mathbb{R}^{m \times m}$ with a normal distribution. A symmetric positive definite $B \in \mathbb{R}^{m \times m}$ is formed by

$$B = L_0 L_0^T + \delta I_m, \qquad \delta > 0, \tag{2.2}$$

where I_m is an identity matrix of order m. Clearly, B is symmetric. The matrix $L_0 L_0^T$ is positive semi-definite since for any non-zero vector $\mathbf{x}$

$$\mathbf{x}^T (L_0 L_0^T)\mathbf{x} = (L_0^T \mathbf{x})^T (L_0^T \mathbf{x}) = \|L_0^T \mathbf{x}\|^2 \geq \mathbf{0}. \tag{2.3}$$

However, $L_0 L_0^T$ may not be strictly positive definite if L_0 is singular. The term δI_m ensures strict positve definiteness by adding δ to its diagonals, thereby shifting all eigenvalues by δ. If $\delta > 0$, then all eigenvalues of B will be strictly positive, ensuring B is positive definite. This guarantees that we can compute the Cholesky factorization of B without any numerical issues.

Another important thing to note is that, δ can be used to control the conditioning of B. We recall from section (1.3.3), that the condition number of B when B is symmetric, is defined as:

$$\kappa(B) = \frac{\lambda_{\max}(B)}{\lambda_{\min}(B)} \tag{2.4}$$

where $\lambda_{\max}(B)$ and $\lambda_{\min}(B)$ are the largest and smallest eigenvalues of B, respectively. In general, B is usually ill-conditioned with a very large condition number so that if δ is large, the process of adding δI_m can regularize the condition number of B, making B well-conditioned, since that will equate to increasing $\lambda_{\min}(B)$. If δ is small, B can still be ill-conditioned but not in an astronomical way. Hence, δ is a hyperparameter we can use to control the condition of B. In this experiment, we choose $\delta = 10^1$, which gives a condition number of $\kappa(B) = 8.09 \times 10^2$.

Since B is symmetric and positive definite, we can compute its Cholesky factorization $B = LL^T$ and construct A using a congruence transformation

$$A = LCL^T \tag{2.5}$$

So that the generalized eigenvalues $\Lambda(A, B)$ is equal to the eigenvalues of the diagonal matrix D. This can be summarized by the following lemma:

Lemma 2.3.1. *Let $A - \lambda B$ be a pencil, where A and B are symmetric, and B is strictly positive definite. Let D be a diagonal matrix and C be unitarily similar to D. Assuming (2.5) holds, then the generalized eigenvalues in $\Lambda(A, B)$ are the diagonal elements of ~~is similar to~~ D*

Proof. Given the generalized problem

$$A\mathbf{v} = \lambda B\mathbf{v}, \qquad \mathbf{v} \neq \mathbf{0} \tag{2.6}$$

Since B is positive definite, then clearly, it is invertible and the generalized eigenvalues $\Lambda(A, B)$ will be the eigenvalues of $B^{-1}A$.

Now

$$\begin{aligned} B^{-1}A &= (LL^T)^{-1}(LCL^T) \\ &= L^{-T}L^{-1}LQDQ^TL^T \\ &= (L^{-T}Q)D(Q^{-1}L^T) \\ &= (L^{-T}Q)D(L^{-T}Q)^{-1} \end{aligned}$$

Therefore $B^{-1}A$ is similar to D~~and hence A(A, B) is similar to D~~[1]. □

The pseudocode for generating A and B is described in Algorithm 3. With the problem

Algorithm 3 Setting up a GEP

Require: D: diagonal matrix with known eigenvalues, δ: regularization hyperparameter

function GENERATE_MATRIX(D, δ)
    Set $m =$ `size`(D)
    Q, _ = `qr`(random.randn(m, m))
    $C = QDQ^T$
    $L_0 =$ `tril`(`random.randn`(m, m))
    $B = (L_0L_0^T) + \delta I$
    $L =$ `cholesky`(B)
    $A = LCL^T$
    return (A, B)
end function

setup completed, and the algorithm described, in the next chapter, we shall discuss the results obtained in these experiments.

[1] **[1]:** remove

CHAPTER 3
EXPERIMENTAL RESULTS AND DISCUSSION

In this chapter, we will present a comprehensive analysis of the experimental results obtained from our implementation of the Spectral Transformation Lanczos algorithm for the symmetric definite generalized eigenvalue problem. We examine the algorithm performance on a test matrix, analyze the effects of ill-conditioning on convergence and accuracy, and compare the error bounds with what was predicted with direct methods.

3.1 Software and Computational Environment

The numerical experiments in this thesis are performed using the Python programming language together with the NumPy 2.0.2 and SciPy 1.13.1 libaries which makes function calls to optimized and efficient LAPACK and BLAS routines for linear algebra computations. These libraries ensure high-performance matrix operations and numerical stability. All computations are performed in **double precision** (64 bit floating point, `float64`) to maintain numerical accuracy and consistency.

For reproducibility, all code is written in Python 3.9.6 and executed within a controlled environment using `virtualenv`. All numerical results have been validated by comparing different levels of precision where applicable and verifying consistency with analytical results when available. Code for the experiments is managed using version control with Git to ensure reproducibility and can be found in https://github.com/AyobamiAdebesin/ayobami_thesis

3.2 Experimental Setup

To evaluate the ST-Lanczos algorithm, we employ Algorithm 3 to generate test matrices A and B with controlled eigenvalue distribution. For the purpose of this thesis, we will be testing with dense matrices. In practice, the ST-Lanczos method is typically applied to sparse matrices. However, we are experimenting with dense matrices, as they are easier to generate and more convenient for testing stability. The eigenvalues are divided into 3 distinct groups, each with a specified range (spread). For each of the three groups, a random set of eigenvalues was generated with a uniform distribution, ensuring that the eigenvalues are distributed evenly within their respective ranges.

- Group 1 contains 1000 eigenvalues in the range $(10^{-3}, 10^{1})$
- Group 2 contains 1000 eigenvalues in the range $(10^{1}, 10^{3})$
- Group 3 contains 1000 eigenvalues in the range $(10^{3}, 10^{7})$

The three sets of eigenvalues are then concatenated into a single array $D \in \mathbb{R}^{3000\times3000}$, which is then used with a regularization hyperparameter $\delta = 10^{1}$, to generate A and B of size 3000×3000. Our shift is choosen to be $\sigma = 1.5 \times 10^{3}$. For this choice of δ, the condition number of A and B are as follows:

$$\kappa(A) = 5.96 \times 10^{11}, \qquad \kappa(B) = 8.09 \times 10^{2}$$

As discussed in Section 2.3, A and B will be symmetric with B being positive definite. The matrix B is factored using the SciPy `cholesky` method which calls LAPACK `xPOTRF`. We

chose to use this since B was designed to be strictly positive definite, which does not require us to use pivoted Cholesky factorization. If it were merely semidefinite, pivoting would be required. We run Algorithm 2 for $n = 2000$ iterations for the spectral problem

$$C_b^T(A - \sigma B)^{-1}C_b\mathbf{u} = \theta\mathbf{u}, \qquad \mathbf{u} \neq \mathbf{0} \tag{3.1}$$

to compute the lanczos decomposition using various decompositions techniques for $A - \sigma B$ that was discussed in Section 2.1. We explore these techniques, discuss the expected residual bounds and discuss the results in the following sections.

3.3 Residual and Error Analysis

To evaluate the efficiency of the ST-Lanczos algorithm for the given problem, we define some metrics and also review some of the residual bounds theorems given in [4].

Definition 3.3.1 (Lanczos Decomposition Residual)**.** *Let equation (2.1) be the decomposition obtained from the completion of Algorithm 2. Then we define the relative residual as:*

$$\textit{Relative Decomposition Residual} = \frac{\|C_b^T(A - \sigma B)^{-1}C_bQ_n - Q_nT_n - \delta_n\mathbf{q}_{n+1}\mathbf{e}_n^T\|}{\|C_b^T(A - \sigma B)^{-1}C_b\|}, \tag{3.2}$$

This residual measures how well the Lanczos algorithm has approximated the eigenvalues of B through the tridiagonal matrix T_n.

Definition 3.3.2 (Generalized Relative Residual)**.** *Let (α_i, β_i) be the generalized eigenvalues of A and B, and $\mathbf{v}_i \neq \mathbf{0}$ be the corresponding computed eigenvector obtained from computing the eigenvalues of T_n, such that $\mathbf{r}_i = (\beta_i A - \alpha_i B)\mathbf{v}_i$. Then we define the relative residual for*

each $i = 1, 2, \ldots n$ *as:*

$$\|\tilde{\mathbf{r}}_i\| = \frac{\|(\beta_i A - \alpha_i B)\mathbf{v}_i\|}{(|\beta_i|\|A\| - |\alpha_i|\|B\|)\|\mathbf{v}_i\|} \tag{3.3}$$

[1]

Definition 3.3.3 (Spectral Transformed Residual)**.** *Let* $(\theta_i, \mathbf{u}_i)$ *be the Ritz-pair obtained from computing the eigenvalues and eigenvectors of* T_n*. Then we define the relative residual for each* $i = 1, 2, \ldots n$ *as:*

$$ST\ Relative\ Residual = \frac{\|C_b^T(A - \sigma B)^{-1}C_b\mathbf{u}_i - \theta_i\mathbf{u}_i\|}{(\|C_b^T(A - \sigma B)^{-1}C_b\| + |\theta_i|)\|\mathbf{u}_i\|} \tag{3.4}$$

Definition 3.3.4 (Best Residuals)**.** *Let* (α_i, β_i) *be the computed generalized eigenvalues of* A *and* B*. We define the "best relative residuals" for* (α_i, β_i) *as*

$$\|\tilde{\mathbf{r}}_i\| = \frac{\sigma_m(\beta_i A - \alpha_i B)}{(|\beta_i|\|A\| + |\alpha_i|\|B\|)} \tag{3.5}$$

where σ_m *is the smallest singular value of the matrix pencil* $(\beta_i A - \alpha_i B)$*.*

This residual assess the quality of the computed eigenvalues independently of the eigenvectors. The smallest singular value is used to determine the "**best achievable residual**" for the eigenvalue pair (α_i, β_i), even with an idealized eigenvector.

We reference the following theorem(s) and lemma(s) from [4], which gives the residual bounds for computed eigenvalues and eigenvectors using the direct method employed in the paper. We will be using these bounds to evaluate the efficiency and stability of the ST-Lanczos algorithm used in this thesis and validate our computed residuals with these bounds.

[1] **[2]:** You need $(|\beta_i|\|A\| + |\alpha_i|\|B\|)$ in the denominator. (add instead of subtract.)

For context we define

$$\eta = \frac{\|A - \sigma B\|^{1/2}}{\|B\|^{1/2}}, \qquad X = C_a^{-1} C_b, \qquad \text{and} \qquad \mu = \frac{\|X\|}{\|X^T D_a X\|}.$$

It is shown in [4] that

$$\eta^2 \|X\|_2^2 \leq \left(1 + \frac{1}{\sigma_0}\right) \frac{\mu}{\min_i \left|1 - \frac{\lambda_i}{\sigma}\right|}$$

where the minimum is taken over all generalized eigenvalues λ_i and σ_0 is as in (1.2). Hence, if μ is not large, if σ is not too close to a generalized eigenvalue, and if σ_0 is not too small, then $\eta^2 \|X\|_2^2$ is not large. These conditions are easily satisfied in practice, so that the presence of $\eta^2 \|X\|^2$ in error bounds poses no problem. It was also shown in [4] that if the spectral transformation matrix is computed explicitly and then its eigenvalue decomposition is then computed, we obtain

$$(C_b + F_1)^T (A + E - \sigma(B + F))^{-1} (C_b + F_1) = \tilde{U}(\Theta + G)\tilde{U}^T, \tag{3.6}$$

$$B + F = (C_b + F_1)^T (C_b + F_1), \tag{3.7}$$

where Θ is the diagonal matrix of computed eigenvalues and $\tilde{U}$ is an exactly orthogonal matrix that is close to the matrix of computed eigenvectors for the spectral transformation matrix $C_b^T (A - \sigma B)^{-1} C_b$~~$C_b^T (A - \sigma B) C_b$~~. The error matrices E, F, and G can be suitably bounded in terms of the unit roundoff. From these error relations it follows that the computed eigenvalues in Θ are close to the eigenvalues of the matrix obtained from a spectral transformation of matrices $A + E$ and $B + F$ that are close to A and B. These error bounds also have consequences for the residuals associated with the computed solution to

the generalized eigenvalue problem.

Theorem 3.3.1 (Residual Bounds for Eigenvalues)**.** *Let C_a, C_b, X, η, and θ_i be computed quantities using the direct method algorithm described in [4]. Let E, F, and F_1 be as in (3.6) and (3.7). In what follows e_n and f_n are modestly growing functions of n more precisely described in [4]. Suppose that $(A+E)-\sigma(B+F)$ is invertible. Without loss of generality, assume that the computed eigenvalues θ_i are in nondecreasing order and let $\hat{\theta}_i$ and $\hat{\mathbf{u}}_i$ for $i = 1, 2, \ldots, r$ be eigenvalues and eigenvectors of*

$$\hat{W} = (C_b + F_1)^T (A + E - \sigma(B + F))^{-1}(C_b + F_1)$$

with the $\hat{\theta}_i$ also in nondecreasing order. Assume that $(C_b + F_1)\hat{\mathbf{u}}_i \neq \mathbf{0}$ and define

$$\hat{\mathbf{v}}_i = (A + E - \sigma(B + F))^{-1}(C_b + F_1)\hat{\mathbf{u}}_i \neq \mathbf{0}.$$

Then

$$\|(\theta_i(A+E) - (1+\sigma\theta_i)(B+F))\hat{\mathbf{v}}_i\|_2 \leq$$

$$u(e_n + f_n)(1 + |\sigma_0|)\eta^2\|X\|_2^2 \left(|\theta_i|\|A+E\|_2 + |1+\sigma\theta_i|\|B+F\|_2\right)\|\hat{\mathbf{v}}_i\|_2 + O(u^2).$$

where $\hat{\mathbf{v}}_i$ is the exact eigenvector of the perturbed problem, and not a computed eigenvector using the Lanczos procedure.

The implication of this theorem is if neither σ_0 nor $\eta\|X\|$ are large, then there exists an approximate eigenvector for which the computed eigenvalues achieve a small residual, although this eigenvector is not actually the computed eigenvector. Thus, the computation

of the eigenvalues on their own is stable. What is important for us to remember here is that choosing the shift σ to avoid extreme proximity to eigenvalues can be shown to give reasonable bounds on $\eta\|X\|_2$. That is, for eigenvalues $\lambda_i = (1+\sigma\theta_i)/\theta_i$, the residuals will remain small if σ is not too close to λ_i.

The situation for the computed eigenvectors is not as satisfactory.

Theorem 3.3.2. *Assume that $\sigma \neq 0$ and $\lambda_i \neq 0$. Under the assumptions of Theorem 3.3.1, we have*

$$\left\|\left(\theta_i A - (1+\sigma\theta_i)B\right)\mathbf{v}_i\right\|_2 \leq |1+\sigma\theta_i| \cdot |1-\sigma/\lambda_i| \left[uc_n + O(u)\left(\eta\|X\|_2 + \left(1+\max(\gamma,1)\left(1+\left|1-\frac{\lambda_i}{\sigma}\right|\right)\right)\eta^2\|X\|_2^2 \right)\right] \|B\|_2\|\mathbf{v}_i\|_2,$$

and

$$\left\|\left(\theta_i A - (1+\sigma\theta_i)B\right)\mathbf{v}_i\right\|_2 \leq |\theta_i| \cdot |1-\lambda_i/\sigma| \cdot |\sigma_0| \left[uc_n + O(u)\left(\eta\|X\|_2 + \left(1+\max(\gamma,1)\left(1+\left|1-\frac{\lambda_i}{\sigma}\right|\right)\right)\eta^2\|X\|_2^2 \right)\right] \|A\|_2\|\mathbf{v}_i\|_2,$$

where $\gamma = \|A\|_2/\|A-\sigma B\|_2$ and c_n does not grow quickly in n.

Theorem 3.3.2 covers two distinct cases. The first case is of primary interest when σ_0

is large. Again, assuming the σ is not so close to an eigenvalue that $\eta\|X\|$ becomes large, in~~In~~ this case the residuals scale with $|1-\sigma/\lambda_i|$ and $|1-\lambda_i/\sigma|$, ensuring accuracy when σ aligns somewhat closely with λ_i but suggesting a loss of accuracy if λ_i is either much larger or smaller than σ. The second case includes $|\sigma_0|$ as a factor in the bounds and is more useful when the scaled shift is not too large. In this case, the residuals scale with only with $|1-\lambda_i/\sigma|$, so that we do not expect a loss of accuracy when λ_i is much smaller than σ_0.

So the key takeaway useful for this thesis is, in order to ensure small residuals and stability, the shift σ should not be too close to any generalized eigenvalue and should not be too large in magnitude. A moderate shift will guarantee small residuals for eigenvalues not much larger than σ but a large shift will result in having small residuals only for eigenvalues near σ. The question that we will resolve in the experiments is whether this pattern holds for the Lanczos algorithm and not just for the direct method from [4].

3.4 LU decomposition

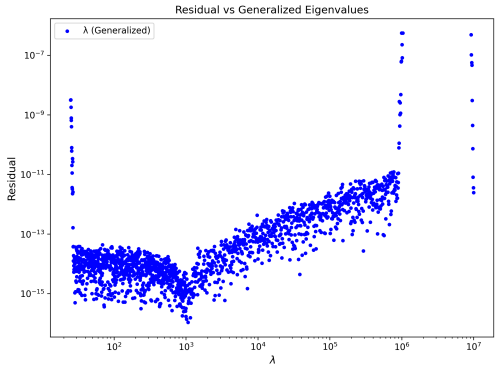
The LU decomposition of $A-\sigma B$ is performed using the `linalg.lu_factor` function in SciPy, which employs partial pivoting. With a moderately large shift $\sigma = 1.5\times 10^3$ not close to a generalized eigenvalue and for $n = 2000$ iterations, the Krylov solution subspace has a dimension of 2000, and the ST Lanczos algorithm achieves a decomposition residual, computed with (3.2), to the order of 10^{-11}, despite a full reorthogonalization. Using a tolerance of 10^{-10} for the Ritz pair residuals computed with (3.4), approximately 78% of the Ritz pairs converge to within an accuracy of order 10^{-14}. The plot of this residuals are

shown to the right in Fig 3.1(b).

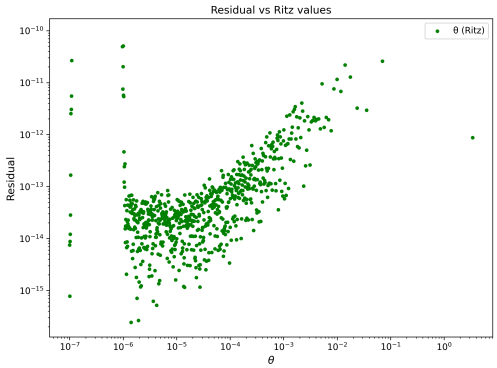
On the other hand, the generalized residuals for these converged Ritz pairs are close to the order of machine precision $u = 10^{-15}$ for generalized eigenvalues not much larger than the shift. As shown in Fig 3.1(a), the residuals are small for eigenvalues that are close to or smaller in magnitude than the shift and gradually tends to increase for larger eigenvalues, consistent with the error bounds stated in Theorem 3.3.2 for a direct method, indicating that the residuals scale with $|1 - \lambda_i/\sigma|$. This scale factor appears in one of the error bounds in Theorem 3.3.2. The third plot, Fig 3.1(c) shows the best possible relative residuals for some choice of $\mathbf{v}_i$. This is the best residuals described in (3.3.4). This shows that every computed eigenvalue can result in a small residual, validating the bound given in Theorem 3.3.1.

For a shift that is large in magnitude, we consider $\sigma = 1.5 \times 10^5$. Running the algorithm with the same parameters as described for the moderate shift, it is observed that the computed eigenvalues and eigenvectors delivers small residuals for eigenvalues that are not too much larger or smaller in magnitude than σ. The plot of for this residual is shown in Fig 3.2(a) shows that for eigenvalues that are orders of magnitude smaller than the shift, the residuals are much smaller. Although, this might not be evident in the plot, but the fact that these residuals are not present implies that the Ritz values for these eigenvalues did not converge in the spectral problem.

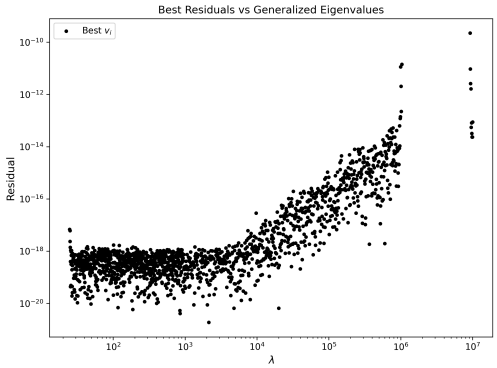
Figure 3.1: Residuals plot with moderate shift $\sigma = 1.5 \times 10^3$



(a)



(b)



(c)

3.5 Eigenvalue Decomposition

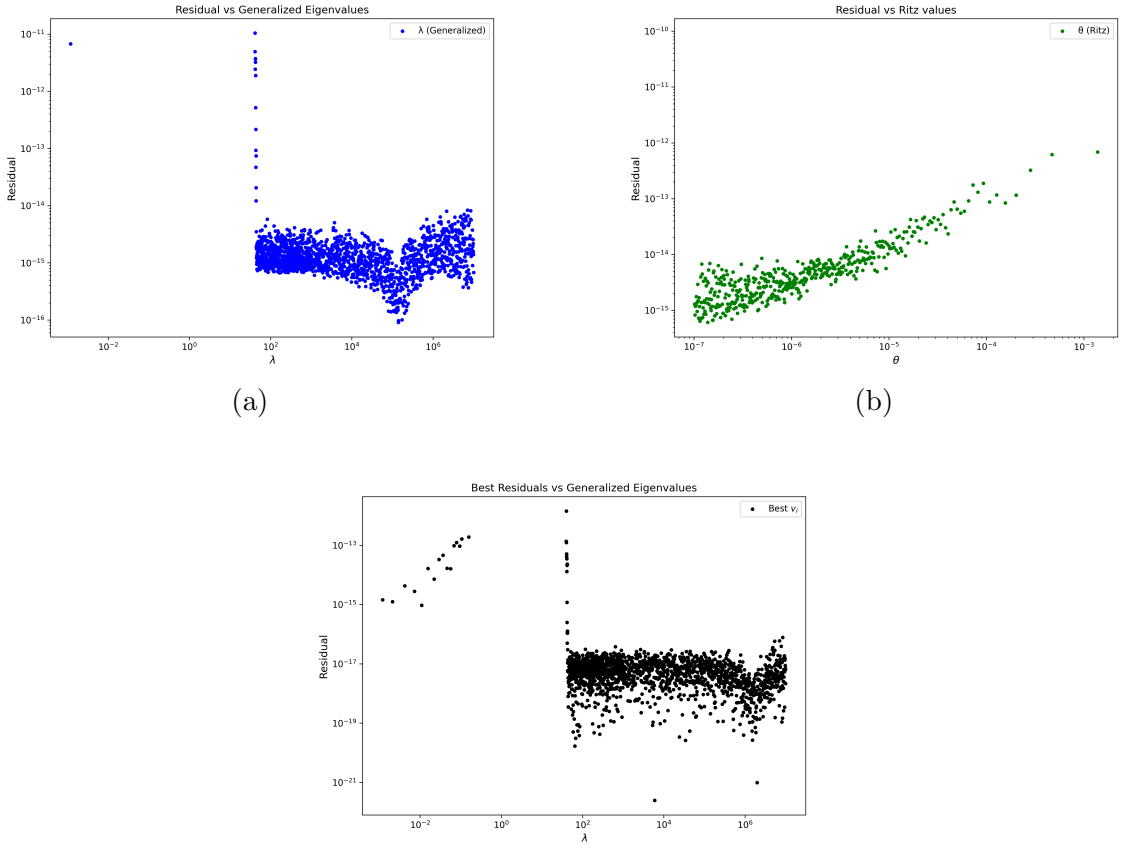
Another decomposition technique we employed is the symmetric eigenvalue decomposition of $A - \sigma B$ given by

$$A - \sigma B = W D W^T, \tag{3.8}$$

so that the spectral transformed problem is given by

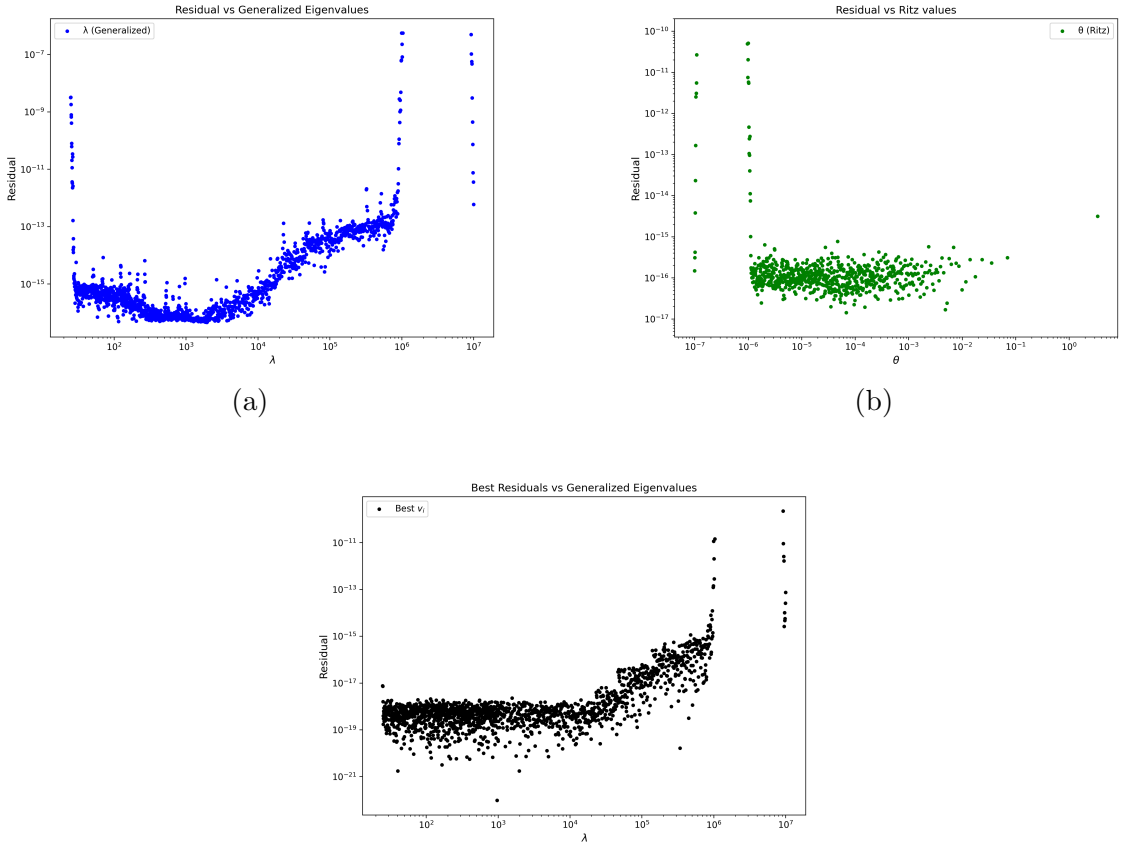
$$C_b^T W^{-T} D^{-1} W^{-1} C_b \mathbf{u} = \theta \mathbf{u}, \qquad \mathbf{u} \neq \mathbf{0} \tag{3.9}$$

Figure 3.2: Residuals plot with large shift $\sigma = 1.5 \times 10^5$



This decomposition, done using `linalg.eigh` function in SciPy, uses the LAPACK `dsyevd` routine for real symmetric matrices, which in turn uses divide-and-conquer algorithms for efficiency. For a moderate shift, $\sigma = 1.5 \times 10$, the Lanczos decomposition residual was observed to be of the order 10^{-29}, indicating a highly accurate decomposition. In Fig 3.3(a), the residuals are close to the order of machine precision $u = 10^{-15}$ for generalized eigenvalues not much larger than the shift. Similar to the LU decomposition, the eigenvalue decomposition achieves small residuals for eigenvalues that are close to or smaller in magnitude than the shift and gradually tends to increase for larger eigenvalues, consistent with the error bounds

Figure 3.3: Residuals plot with moderate shift $\sigma = 1.5 \times 10^3$



stated in Theorem 3.3.2. One the other hand, 78% of the Ritz values converged to within an accuracy of machine precision. As shown in Fig Fig 3.3(b), there is no apparent trend of increasing residuals for values larger than the shift. This maybe attributed to the use of a symmetric decomposition. Fig 3.3(c) also shows the best possible relative residuals for some choice of $\mathbf{v}_i$. This shows that every computed eigenvalue can result in a small residual, validating the bound given in Theorem 3.3.1.

CHAPTER 4
CONCLUSION

This thesis has investigated the application and performance of the Spectral Transformation Lanczos algorithm for solving symmetric definite dense generalized eigenvalue problem. Through the numerical experiments, we validated our results with proven error bounds in direct methods, considered the implication of several factorization techniques, and the impact of certain properties of the matrix on the accuracy of the results. In this concluding chapter, we summarize our key findings, discuss the broader implications of this work, acknowledge limitations, and outline promising directions for future research.

4.1 Summary of Key Findings

The experiments in this thesis have uncovered some interesting results regarding the spectral transformation lanczos algorithm for generalized eigenvalue problems. First, we have established that moderate shift σ that is not too close to a generalized eigenvalue will guarantee small residuals for eigenvalues not much larger than σ, but a large shift will result in having small residuals only for eigenvalues near σ.

Secondly, our analysis of the eigenvalue sensitivity revealed the relationship between the conditioning of the matrices, and the accuracy of computed eigenvalues for various factorizations of the shifted matrix $A-\sigma B$. We observed that for any factorization involving symmetry (eigenvalue decomposition or LDL^T factorization), the ST-Lanczos appears to be stable and the Ritz pairs converged to the order of unit round off u for the n lanczos steps. The generalized eigenvalues also converged, achieving unit round off for all computed

eigenvalues closer and farther away from the shift. This poses an interesting question: "Can we prove stability for any symmetric decomposition of $A - \sigma B$"?

Finally, for the LU decomposition of $A - \sigma B$, we observe that the lanczos procedure was stable but the behavior is largely dependent on the conditioning of $A - \sigma B$. However, our results indicated that the generalized residuals were insensitive to the conditioning of the problem.

4.2 Importance and Implications

From a theoretical perspective, this work advances our knowledge of spectral transformation, matrix conditioning and eigenvalue sensitivity in the context of generalized eigenvalue problems. Our results showed that the bounds for direct methods, appear to hold true for iterative methods. This work goes a step further at highlighting an interesting property of spectral transformation methods that can determine stability for such methods, both in the direct and iterative context. This contributes to the broader field of numerical linear algebra by providing a more comprehensive framework for analyzing iterative eigenvalue solvers.

By characterizing the relationship between matrix factorizations and algorithm convergence, we have developed a better understanding of how spectral transformations affect the convergence of properties of Krylov subspace methods.

REFERENCES

[1] T. Ericsson and A. Ruhe, *The spectral transformation lánczos method for the numerical solution of large sparse generalized symmetric eigenvalue problems*, Mathematics of Computation, 35 (1980), pp. 1251–1268.

[2] G. H. Golub and C. F. Van Loan, *Matrix Computations - 4th Edition*, Johns Hopkins University Press, Philadelphia, PA, 2013.

[3] C. B. Moler and G. W. Stewart, *An algorithm for generalized matrix eigenvalue problems*, SIAM Journal on Numerical Analysis, 10 (1973), pp. 241–256.

[4] M. Stewart, *Spectral transformation for the dense symmetric semidefinite generalized eigenvalue problem*, 2024.

[5] L. N. Trefethen and D. Bau, *Numerical Linear Algebra, Twenty-fifth Anniversary Edition*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2022.