The Spectral Transformation Lanczos Algorithm for the Symmetric-Definite Generalized Eigenvalue Problem: A Comparative Analysis with Conditioning Insights~~Spectral Transformation Lanczos Algorithm for Symmetric-Definite Generalized Eigenvalue Problems: 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 (ST-Lanczos) algorithm to a dense symmetric-definite generalized eigenvalue problem involving real, symmetric matrices A and B, with B being positive definite and possibly~~,~~[1] ill-conditioned. The Lanczos algorithm is a well-known iterative algorithm for computing the eigenvalues of a symmetric matrix and it works well finding extreme points in the spectrum.~~if the spectrum of the eigenvalues are well-spaced.~~ 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 in using ST-Lanczos for sparse problems, we adapt this technique to dense problems and analyze how the error bounds already proven for direct methods plays out in an iterative context. [2]

This study primarily focuses 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~~lanczos~~ algorithm, Ritz~~ritz~~ values, Krylov~~krylov~~ subspaces, spectral transformation, orthogonality

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

[2] **[2]:** Mostly we are testing on dense problems. I would not say we are adapting it to dense matrices. We simply happen to use some dense test problems because it is convenient.

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

by

Ayobami Adebesin

Committee Chair: Michael Stewart

Committee: Russell Jeter~~Jeter Russell~~

Vladimir Bondarenko

~~Alvin Das~~[3]

~~Albert Einstein~~[4]

Electronic Version Approved:

Office of Graduate Studies

College of Arts and Sciences

Georgia State University

May 2025

DEDICATION

I dedicate this project to God.

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.[5]

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~~have~~ 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~~Jeter Russell~~, 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 —[6] 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

[5] **[5]:** You used \\[5pt] at the end of this paragraph. Please just separate paragraphs with blank lines here and in the rest of the thesis. In general \\is best avoided outside of tabular or array environments. It sort of works, but it's not a normal way to mark a paragraph in LaTeX and it can cause worse output with underfull hboxes. I don't see any reason for the [5pt] either, so I'd also remove that. Using \par is also not recommended. Finally, it's good practice to put some blank lines around the \section, etc. It makes them easier to spot in the LaTeX code.

[6] **[6]:** Use a textemdash here instead of -.

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. [7]

[7] **[7]:** This was all a very nice acknowledgment.

TABLE OF CONTENTS

ACKNOWLEDGMENTS . . . . **v**

LIST OF TABLES . . . . **ix**

LIST OF FIGURES . . . . **x**

1 INTRODUCTION **2**

1.1 Background 2

1.2 Mathematical Preliminaries 3

1.2.1 Notation 4

1.2.2 Floating Point Arithmetic 4

1.2.3 Vector Norms 5

1.2.4 General Matrix Norms 6

1.2.5 Induced Matrix Norms 7

1.2.6 Conditioning and Stability 8

1.2.7 Congruence Transformation 9

1.2.8 The Standard Eigenvalue Problem 10

1.2.9 The Generalized Eigenvalue Problem 11

1.2.10Lanczos Algorithm 13

1.2.11Spectral Transformation 14

1.3 Problem Discussion 16

1.4 Numerical Experiments 17

1.5 Motivation of Study 18

1.6 Significance of Study 19

2 LITERATURE REVIEW **20**

3 METHODOLOGY AND ALGORITHM DESCRIPTION **22**

3.1 Spectral Transformation . . . 22

3.2 Lanczos decomposition . . . 26

3.3 Experimental Setup . . . 29

4 EXPERIMENTAL RESULTS AND DISCUSSION . . . **33**

5 CONCLUSION . . . **34**

5.1 Summary of Key Findings . . . 34

5.2 Importance and Implications . . . 35

5.2.1 Theoretical Contributions . . . 35

5.2.2 Practical Implications . . . 36

Appendices . . . 36

A Something . . . 37

B Something Else . . . 38

REFERENCES . . . **39**

LIST OF TABLES

LIST OF FIGURES

[8]

---

[8] **[8]:** On your committee list I suggest adding (\qquad) twice to create two extra blank committee members. The thesis class seems to want to add a fixed number of members and then fills in some default names.

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 matrices. 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[1] (GEP). The GEP takes the form $A\mathbf{v} = \lambda B\mathbf{v}$$\cancel{Av = \lambda Bv}$ where A and B are square matrices, λ is a generalized eigenvalue, and $\mathbf{v} \neq \mathbf{0}$$\cancel{v}$ 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 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] **[9]:** use a space here

1.2.1 Notation

Throughout this study, we make use of the following notations: [2]

$A \in \mathbb{C}^{m\times n}$: denotes a matrix~~square or rectangular matrices~~

~~$Q \in \mathbb{C}^{m\times m}$: denotes unitary or orthogonal matrices~~[3]

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

$\mathbf{x} \in \mathbb{C}^m$: denotes a column vector~~s~~[4]

~~Greek letters α, β... : denotes scalars in $\mathbb{C}$~~[5]

A^T : denotes the transpose of matrix A

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

$\otimes$: denotes the Kronecker product of two matrices

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

$A^{(k)}$: denotes the matrix A at the kth~~kth~~ step of an iteration

1.2.2 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

$$x = \pm(m/\beta^t)\beta^e,$$ [6]

[2] **[10]:** I would not try to use define letters like Q to mean specific things here. I'd also not use the convention that all scalars should be Greek letters. That's very hard to stick to consistently.

[7] 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 $x \in$ F is unique, we can restrict the range of **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 $fl : \mathbb{R} \rightarrow \mathbf{F}$ 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 1.2.1. *If $x \in \mathbb{R}$ is in the range of* **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.

1.2.3 Vector Norms

[8] Norms are generally used to capture the notions of size and distance in a vector space. A norm is a function $\|\cdot\| : \mathbb{C}^m \rightarrow \mathbb{R}$ satisfying the following properties for all vectors **x** and **y**

[7] **[15]:** You need to add punctuation to the end of displayed equations. Before a where, a comma works. If an equation ends a sentence you need a period. I might mark a few more, but there are a lot of equations that need punctuation.

[8] **[16]:** I don't think you need to review standard linear algebra like basic properties of vector and matrix norms. I would remove this section as well as the one on matrix norms. It should suffice to say what norm you are using when you first use a norm.

and scalars $\alpha \in \mathbb{C}$:

$$(1)\ \|\mathbf{x}\| \geq 0,\ \text{and } \|\mathbf{x}\| = 0 \text{ only if } \mathbf{x} = 0,$$

$$(2)\ \|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|,$$

$$(3)\ \|\alpha\mathbf{x}\| = |\alpha|\|\mathbf{x}\|$$

The most important class of vector norms are the p-norms and are defined as follows:

$$\begin{aligned}
\|\mathbf{x}\|_1 &= \sum_{i=1}^{m} |x_i| = 1, \\
\|\mathbf{x}\|_2 &= \Big(\sum_{i=1}^{m} |x_i|^2\Big)^{1/2}, \\
\|\mathbf{x}\|_\infty &= \max_{1\leq i\leq m} |x_i|, \\
\|\mathbf{x}\|_p &= \Big(\sum_{i=1}^{m} |x_i|^p\Big)^{1/p}, \qquad (1 \leq p < \infty)
\end{aligned}$$

1.2.4 General Matrix Norms

Similar to a vector norm, a matrix norm is a function $\|\cdot\| : \mathbb{C}^{m\times n} \to \mathbb{R}$ satisfying the following properties for all matrices A and B and scalars $\alpha \in \mathbb{C}$:

$$(1)\ \|A\| \geq 0,\ \text{and } \|A\| = 0 \text{ only if } A = 0,$$

$$(2)\ \|A + B\| \leq \|A\| + \|B\|,$$

$$(3)\ \|\alpha A\| = |\alpha|\|A\|$$

The simplest and most important example of a general matrix norm is the Frobenius norm

$$\|A\|_F = \left(\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2 \right)^{1/2} \tag{1.1}$$

[9] Let a_j be the jth column of A, equation 1.1 can be written as

$$\|A\|_F = \left(\sum_{j=1}^{n} \|a_j\|_2^2 \right)^{1/2} \tag{1.2}$$

In a more compact form ,we can rewrite it as

$$\|A\|_F = \sqrt{tr(A^*A)} = \sqrt{tr(AA^*)} \tag{1.3}$$

where $\mathrm{tr}(A)$ denotes the trace of A, which is the sum of its diagonal entries.

1.2.5 Induced Matrix Norms

Another important class of matrix norm is the *induced matrix norms.* These are matrix norms induced by vector norms, defined in terms of the behaviour of a matrix as an operator between its normed domain and range spaces.

Let $A \in \mathbb{C}^{m\times n}$ be a matrix with vector norms $\|\cdot\|_{(n)}$ and $\|\cdot\|_{(m)}$ on the domain and the range of A, respectively, the induced matrix norm $\|A\|_{(m,n)}$ is defined as:

$$\|A\|_{(m,n)} = \sup_{\substack{\mathbf{x}\in\mathbb{C}^n \\ \mathbf{x}\neq 0}} \frac{\|A\mathbf{x}\|_{(m)}}{\|\mathbf{x}\|_{(n)}} = \sup_{\substack{\mathbf{x}\in\mathbb{C}^n \\ \|\mathbf{x}\|_{(n)}=1}} \|A\mathbf{x}\|_{(m)} \tag{1.4}$$

We can think of the induced matrix norm as the maximum factor by which A can stretch a vector.

[9]**[17]:** I suggest removing this section, but this seems like a good place to say that I would suggest you avoid \bigg and similar things. (Although they can be useful if you split equations across lines.) It is usually easier and better just to let LaTeX size parenthesis, or other delimiters, automatically with \left(and \right). You also get a little bit of error checking from LaTeX if you use left and right.

The following matrix norms are useful:

- 1-norm: $\|A\|_1 = \max_{1\leq j\leq n} \|\sum_{i=1}^{m} a_{ij}\|_1$, maximum column sum.

- ∞-norm: $\|A\|_\infty = \max_{1\leq i\leq m} \|\sum_{j=1}^{n} a_{ij}\|$, maximum, row sum.

- 2-norm $= \sqrt{\lambda_{\max}(A^T A)}$, square root of the largest eigenvalue of $A^T A$.

The Frobenius norm and the 2-norm have many special properties, one of which is invariant under unitary multiplication. That is for an orthogonal or unitary matrix Q,

$$\|QA\|_2 = \|A\|_2, \qquad \|QA\|_F = \|A\|_F \tag{1.5}$$

1.2.6 Conditioning and Stability

[10] Given any mathematical problem $f : X \to Y$, the conditioning of that problem pertains to the perturbation behaviour of the problem, while stability of the problem pertains to the perturbation behaviour of 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

[10]**[18]:** You labeled this `section1.2.6`. By using a labeling convention involving specific numbers, you give up all the main benefit of using labels. You did something similar for labeling equations. The nice thing about labels is that you get correct references even if you change the order of things or add new sections or equations. My own convention is that I use a few letters to denote the type of thing I'm labeling, a colon for a separator, and then a more descriptive name. For this I would have done something like `sec:ConditioningAndStability`, or maybe something a little more abbreviated. For equations, I usually use `eq:` as a prefix to a descriptive label. I would keep this section in some form, but I would add some discussion of backward error and how the condition number determines how the backward error impacts a computed solution.

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 $A \in \mathbb{C}^{m\times n}$, the condition number with respect to a given norm is defined as~~:~~[11]

$$\kappa(A) = \|A\| \cdot \|A\|^{-1} \tag{1.6}$$

In simpler terms, the condition number quantifies how the relative error in the solution of a linear system $Ax = b$ can be amplified when there is a small perturbation in the input vector 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”~~”small”~~ or “large” ~~”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

$$\kappa(A) = \frac{\sigma_1}{\sigma_m} \tag{1.7}$$

where σ_1 and σ_m are the largest and smallest singular values of A respectively.

1.2.7 Congruence Transformation

Let A and B be square matrices. A and B are said to be congruent ($A \sim B$[12]) if there exists an invertible matrix P such that

$$B = P^T AP \tag{1.8}$$

[11] **[19]:** remove
[12] **[20]:** remove

Like, similarity transformation, rank is preserved under congruence. However, eigenvalues are not preserved under congruence transformation.[13]

1.2.8 The Standard Eigenvalue Problem

[14] Let $A \in \mathbb{C}^{m \times m}$ be a square matrix. A nonzero vector $\mathbf{v} \in \mathbb{C}^m$ is said to be an *eigenvector* of A, and $\lambda \in \mathbb{C}$ its corresponding *eigenvalue* if,

$$A\mathbf{v} = \lambda\mathbf{v}, \qquad \mathbf{v} \neq 0 \tag{1.9}$$

The (multi) set of all eigenvalues of A is called the *spectrum* of A and is denoted by $\mathrm{spec}(A)$ ~~$spec(A)$~~.[15] The problem of computing the set of eigenvalues $\lambda \in \mathbb{C}$ and eigenvectors $\mathbf{v} \in \mathbb{C}^m$ that satisfies equation 1.9 is called the *standard eigenvalue problem*.

Equation (1.9) can be written as $(A - \lambda I)\mathbf{v} = 0$. Since $\mathbf{v} \neq 0$, this implies that $A - \lambda I$ is singular. We define the eigenspace, E_λ of A corresponding to an eigenvalue $\lambda \in spec(A)$ as the set of all eigenvectors, together with the zero vector, associated with λ as follows:

$$E_\lambda = \mathcal{N}(A - \lambda I) = \{\mathbf{v} \in \mathbb{C}^m \mid (A - \lambda I)\mathbf{v} = 0\} \tag{1.10}$$

The dimension of this vector space is called the *geometric multiplicity* of $\lambda \in spec(A)$. One of the many ways we can compute the eigenvalues of a matrix is by solving the characteristic

[13] **[21]:** This is much less important to note than the fact that congruence of both A and B for a matrix pencil does preserve generalized eigenvalues. I'd get rid of this as a separate section and merge it with the section on the generalized eigenvalue problem.

[14] **[22]:** As with norms, you can assume your reader knows this. You don't really need this section. If there is notation you define here, you should define it elsewhere when it is first used.

[15] **[23]:** Function names should not be in italics. This is also true for det and tr. An exception might by if you are using a document style in which ordinary text is in italics in theorems or definitions. So the rule is that function names should be in the same style as ordinary text.

polynomial of A defined by

$$p_A(\lambda) = det(A - \lambda I) = 0.$$

The roots of $p_A(\lambda)$ corresponds to the eigenvalues of A. However, using the characteristic polynomial for computing the eigenvalues of a matrix is not considered effective since polynomial root finding is an ill-conditioned problem. In practice, we often use eigenvalue value solvers that are stable and those that exploits the structure of special matrices to compute eigenvalues in a fast and efficient manner. These eigenvalue algorithms are generally categorized into 2 classes - Direct solvers and Iterative solvers.

1.2.9 The Generalized Eigenvalue Problem

Let $A, B \in \mathbb{C}^{m\times m}$~~$A, B \in \mathbb{C}^{mxm}$~~, be any general square matrices. Then the set of all matrices $A - \lambda B$ with $\lambda \in \mathbb{C}$ is called a *pencil.* [16] The *generalized eigenvalues* of $A - \lambda B$ are the elements of the set $\Lambda(A, B)$ defined by

$$\Lambda(A, B) = \{z \in \mathbb{C} : \det(A - zB) = 0\} \tag{1.11}$$

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

[16] **[24]:** I would suggest being a little more precise here. If a pencil is just the set of all matrices $A - \lambda B$ with $\lambda \in \mathbb{C}$, then with $A_1 = 0$ and $B_1 = I$ and $A_2 = I$ and $B_2 = I$, you have defined this so that $A_1 - \lambda B_1$ and $A_2 - \lambda B_2$ are the same pencil. In both cases, the set is just all multiples of I. But these pencils are distinct with different generalized eigenvalues. It would be more correct to say that a pencil is an expression of the form $A - \lambda B$, where A and B are in $\mathbb{C}^{m\times m}$.

$$p_{A,B}(\lambda) = \det(A - \lambda B) = 0 \tag{1.12}$$

[17]

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

$$A\mathbf{v} = \lambda B\mathbf{v} \tag{1.13}$$

then $\mathbf{v}$~~v~~ is a generalized eigenvector of A and B. The problem of finding nontrivial solutions to (1.13) is known as the *generalized eigenvalue problem.*

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

$$B^{-1}A\mathbf{v} = \lambda\mathbf{v} \tag{1.14}$$

In this case, the generalized eigenvalue problem has m eigenvalues if $\mathrm{rank}(B) = m$~~$rank(B) = m$~~. This suggests that the generalized eigenvalues of A and B are equal to the eigenvalues of $B^{-1}A$. If B~~B~~ is singular or rank deficient, then the set of generalized eigenvalues $\Lambda(A, B)$ may be finite, empty or infinite. If the $\Lambda(A, B)$ 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.[18]

If A and B have a common null space, then every choice of λ will be a solution to (1.13).

---

[17] **[25]:** Define regular and singular pencils here.

[18] **[26]:** This part about finite, empty, or infinite is very confusing and I'm not entirely certain I know what you were getting at. The set $\Lambda(A, B)$ is infinite only if the pencil is singular, because then every λ can be considered an eigenvalue. But we probably want to assume the pencil is regular before we get to this point. The singular case has some complications we don't want to get into. I think the point you want to make is that every individual eigenvalue is finite, zero, or infinite, where we define the eigenvectors associated with $\lambda = \infty$ as the null vectors of B. In the regular case, there are exactly m generalized eigenvalues, counting multiplicities.

In this case, we say that the pencil $A - \lambda I$ is *singular*. Otherwise, we say that the pencil is *regular*.~~Such problems are referred to as *ill-disposed* problems.~~ For the purpose of this study, we shall assume that A and B do not have a common null space, that is

$$\mathcal{N}(A) \cap \mathcal{N}(B) = \{\mathbf{0}\} \tag{1.15}$$

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.[19] ~~In addition to that, we shall assume that A and B are dense matrices.~~[20]

1.2.10 Lanczos Algorithm

The Lanczos algorithm is an iterative method in numerical linear algebra used in finding the eigenvalues and eigenvectors of a *hermitian*~~*symmetric*~~[21] 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"~~"most useful"~~ eigenvalues of the matrix —[22] 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

[19] **[27]:** The definition of regular and singular pencils and the assumption of regularity need to be near the beginning of this section, after you define a pencil and generalized eigenvalues, but before you talk about anything else. You might also point out that for a singular pencil, the characteristic polynomial is identically zero and every λ can be considered an "eigenvalue."

[20] **[28]:** remove

[21] **[29]:** I haven't made the change, but you are referring to symmetric matrices, even if they are complex. Either we need to focus on real matrices or change symmetric to hermitian almost everywhere, including in the title of the thesis. I focused on real matrices in my paper.

[22] **[30]:** \textemdash

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 $A \in \mathbb{C}^{m \times m}$, and an initial vector v_1[23], the Lanczos algorithm produces a sequence of vectors $v_1, v_2, \cdots, v_n$ (where n is the number of iterations) that forms an orthonormal basis for the n-dimensional Krylov subspace

$$\mathcal{K}_n(A, v_1) = span(\{v_1, Av_1, A^2v_1, \ldots, A^{n-1}v_1\}) \tag{1.16}$$

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

1.2.11 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 tranformation 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 ~~eigenvalues and eigenvectors are fundamentally tied to matrix operations. By~~[24] 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.~~and thus control which~~

[23] **[31]:** Make this vector and the ones below bold. Write span in a roman font.
[24] **[32]:** remove

~~part of the spectrum, we are interested in.~~ There are various types of spectral transformations[25], but the one that is particualar interest in this thesis is the *shift-invert* transformation. The shift-invert transformation involves tranforming 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~~Assuming~~ 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~~, we~~ can pick a shift $\sigma \in \mathbb{R}$ that is not an eigenvalue of A. The shifted and inverted matrix~~formulation of the problem~~ 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}$$\cancel{\{(\lambda_j - \sigma)^{-1}\}}$, for each eigenvalue λ_j of A.~~where $\{\lambda_j\}$ are the eigenvalues of A.~~ This shifts the spectrum of A, making the eigenvalues near σ much more prominent in the transformed matrix.

For a generalized eigenvalue problem given in (1.13), if we introduce a shift $\sigma \in \mathbb{R}$ so that $A - \sigma B$ is non singular, the shifted and inverted~~shift-invert~~ formulation of the problem is given by

$$(A - \sigma B)^{-1} Bv = \theta v \tag{1.17}$$

[26] where $\theta = 1/(\lambda - \sigma)$.

---

[25] **[33]:** remove

[26] **[34]:** bold vectors.

~~The formulation shifts the spectrum of the generalized eigenvalues $\Lambda(A, B)$ towards σ.~~[27][28] 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 spectrumand, ~~, and~~ by using an iterative method like the Lanczos algorithm, it is likely~~possible~~ that the algorithm will converge quickly to these extreme eigenvalues in the new spectrum.

1.3 Problem Discussion

[29] In this section, we provide a brief but formal statement of the problem we are trying to solve, the methodological approach we used in solving the problem, and discuss the challenges involved in solving these kind of problems.

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

$$A\mathbf{v} = \lambda B\mathbf{v}, \qquad \mathbf{v} \neq 0 \tag{1.18}$$

where A and B are $m \times m$ real symmetric matrices, B positive definite. Both A and B are dense matrices, meaning that a significant proportion of their entries are non-zero.

[27] **[35]:** remove
[28] **[36]:** With the inverse, it's a rational transformation of the spectrum, not a simple shift.
[29] **[37]:** It seems disconnected to have a discussion of the problem here after having a discussion of the generalized eigenvalue problem in an earlier section. I'd move everything into that earlier section and add more material on the symmetric (or hermitian) definite generalized eigenvalue problem. There are several things missing, like the fact that for the symmetric definite problem, the matrices are simultaneously congruent to a diagonal. Then in the previous section on the spectral transformation, you could state the full theorem that I had in my paper relating the eigenvalues and eigenvectors of the original problem to the eigenvalues and eigenvectors of the shifted problem.

The goal is to compute the set of generalized eigenvalues $\Lambda(A, B)$ that satisfy this equation using the ST-Lanczos algorithm. We then proceed by formulating a shift-inverted form of the problem given by equation (1.17), thereby transforming it into a standard eigenvalue problem, which can then be solved using the Lanczos algorithm. 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 how well this method performs of these type of problems, we will setup a well-defined problem by generating synthetic matrices with known eigenvalue distribution, and we will implement the ST-Lanczos algorithm and compare its performance against the direct method based on the paper by Stewart. We will then investigate the relationship between matrix conditioning, shift selection, the accuracy of computed eigenvalues and the sensitvity of the residuals to ill-conditioning.

1.4 Numerical Experiments

[30] The numerical experiments in this thesis are performed using the Python programming language together with the NumPy and SciPy libraries 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 envi-

[30]**[38]:** This material would be better if it were at the beginning of the section where you give results of experiments.

ronment 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

1.5 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 Michael Stewart, 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 dense 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 faster and more efficient that the current ones we have today.

1.6 Significance of Study

[31] The ST-Lanczos algorithms offers the potential for significant computational efficiency compared to direct methods, especially when only a subset of eigenvalues is required. ~~This study aims to optimize the algorithm's performance for dense problems, which could lead to faster and more efficient solutions for large scale eigenvalue computations.~~[32][33]

[31] **[39]:** I would not make something this short into a single section. It might also go better earlier in the paper. It's good to put a general discussion of the problem and your motivation very early in a paper (or thesis).

[32] **[40]:** remove

[33] **[41]:** as noted, we just happen to be using a dense test problem to look at stability.

CHAPTER 2
LITERATURE REVIEW

[1]

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 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 chapter, we discuss some of the research that has been done on this topic.

Golb & Van Loan, 2013 considered the case when B is invertible, in which the problem is reduced to $B^{-1}Av = \lambda v$. 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 $L^{-1}AL^{-T}y = \lambda y$ where $y = L^T v$, which can then be solved by using the symmetric QR algorithm to compute a Schur decomposition.

The QZ algorithm (Moler and Stewart, 1973) for the non-symmetric GEP, is an iterative method that generalized the QR algorithm, to handle singular or ill-conditioned 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,

[1] **[42]:** Literature review is frequently given near the beginning. You also need to have proper citations using labels for references in a bibliography. Names and years are not standard, at least in math.

it is computationally expensive, thereby limiting its use to small or medium sized matrices.

CHAPTER 3
METHODOLOGY AND ALGORITHM DESCRIPTION

3.1 Spectral Transformation

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.

Consider the symmetric-definite generalized eigenvalue problem:

$$A\mathbf{v} = \lambda B\mathbf{v}, \qquad \mathbf{v} \neq 0 \tag{3.1}$$

where A and B are $m \times m$ real, sparse, symmetric and B is positive definite or positive semi-definite.

Problem (3.1) can be reformulated as

$$\beta A\mathbf{v} = \alpha B\mathbf{v}, \qquad \mathbf{v} \neq 0 \tag{3.2}$$

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(3.2) is useful when describing the error bounds, as we shall later see. We shall alternate between (3.1) and (3.2) when convenient. We also observe that the symmetric-definite generalized eigenvalue problem have real eigenvalues.

To compute the eigenvalues and eigenvectors that satisfy equation(3.1) 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.

Let $\sigma \in \mathbb{R}$ be a desired shift such that $A - \sigma B$ is non-singular. The shifted problem takes the form:

$$(A - \sigma B)v = (\lambda - \sigma)Bv \tag{3.3}$$

[1] 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 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 Michael Stewart(2024) and Thomas Ericsson (1960) 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 uses the Bunch-Kaufman pivoting scheme with ”rook pivoting” which is stable. Although the standard LDL^T factorization (without ”rook pivoting”) is

[1] **[43]:** bold vectors

available in 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 factorization is an eigenvalue decomposition of $A - \sigma B$. If we use a symmetric eigenvalue decomposition $A - \sigma B = UDU^T$, 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 the conditioning of A and 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 problems become computationally expensive and not feasible in reality.

Lastly, we can 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 and 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.

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.

Continuing with the algorithm derivation, if we assume $\lambda \neq \infty$ and $\mathbf{v} \neq \mathbf{0}$. Since B is positive definite, Michael Stewart (2024), proved that we can compute a Cholesky factorization $B = C_b C_b^T$, and apply the shift-invert spectral transformation to transform equation(3.1) into its spectral form as described in section (1.2.11) such that $\theta = 1/(\lambda - \sigma)$ is an eigenvalue of the problem :

$$C_b^T (A - \sigma B)^{-1} C_b \mathbf{u} = \theta \mathbf{u}, \qquad \mathbf{u} \neq \mathbf{0} \tag{3.4}$$

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

Conversely, assume that $\mathbf{u} \neq \mathbf{0}$ is an eigenvector of (3.4) and θ its corresponding eigenvalue, then the vector $v = (A - \sigma B)^{-1} C_b \mathbf{u} \neq \mathbf{0}$ is an eigenvector for (3.2), with eigenvalue $(1 + \sigma\theta, \theta)$, provided $C_b \mathbf{u} \neq \mathbf{0}$.

[2] Equation (3.4) gives us the spectral transformed version of the original generalized problem. 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 (3.4) is not desirable in a realistic problem since it does not preserve sparsity and will be very inefficient on most realistic problems.~~as it will make the Lanczos algorithm unstable.~~ ~~Forming the matrix directly also has the disadvantage that the matrix might no longer be symmetric which could prevent the Lanczos algorithm from converging.~~[3][4] The right thing to do is to use the LU for $A - \sigma B$ as explained earlier. This will be explored in the next section.[5]

[2] **[44]:** This material is the main theorem on spectral transformation. I think it belongs in a single section on spectral transformation.

[3] **[45]:** remove

[4] **[46]:** It can be formed in a symmetric way; I did that in my paper for the direct method.

[5] **[47]:** Of course, based on our recent results, you might say that a stable decomposition such as LU could be used, but that we will see that there are some observed stability advantages to decompositions that preserve symmetry.

3.2 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 1.2.10, 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 eigenvalues approximation 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 given as follows:

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 b and set an initial vector $q_1 = b/\|b\|_2$
  Set $\beta_0 = 0$ and $q_0 = 0$
  for $j = 1, 2, \ldots, n$ **do**
    $v = Aq_j$
    $\alpha_j = q_j^T v$
    $v = v - \beta_{j-1}q_{j-1} - \alpha_j q_j$
    Full reorthogonalization: $v = v - \sum_{i\leq j}(q_i^T v)q_i$
    $\beta_j = \|v\|_2$
    if $\beta_j < tol$ **then**
      restart or **exit**
    end if
    $q_{j+1} := v/\beta_j$
  end for
end function

[6] After the completion of algorithm 1, the α's and β's are used to construct the tridiagonal

[6]**[48]:** I believe you used α and β for the generalized eigenvalue represented as a pair (α, β). Under the circumstances, I think γ and δ would be a better choice for Lanczos. There are a lot of unbolded vectors in this section.

matrix $T_n \in \mathbb{R}^{n\times n}$ and the vectors q_j's are stacked together to form an orthogonal matrix $Q_n \in \mathbb{R}^{m\times n}$ given by:

$$T_n = \begin{pmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \beta_2 & \alpha_3 & \beta_3 & \\ & & \ddots & \ddots & \vdots \\ & & & \beta_{n-1} & \alpha_n \end{pmatrix}$$

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

The decomposition is given by

$$AQ_n = Q_n T_n + \beta_n q_{n+1} e_n^T \tag{3.5}$$

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 the Algorithm 1.

Let $\theta_i, i = 1, 2, \ldots n$(which can be computed by standard functions in using any eigenvalue solver) be the eigenvalues of T_n, and $\{y_i\}_{i=1:n}$ be the associated eigenvectors. The $\{\theta_i\}$ are called the *Ritz values* and the vectors $\{Q_n y_i\}_{i=1:n}$ 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 (3.3), Algorithm (1) can be applied to equation (3.3) with some slight modifications. We shall now give the spectral form of Algorithm (1):

Algorithm 2 Spectral Lanczos Algorithm for (3.4)

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

After applying the lanczos procedure to the spectral transformed problem (3.4), we then compute the converged Ritz pairs using a certain tolerance. The converged Ritz pairs are mapped to the generalized eigenvalues and eigenvectors where we can observe the behaviour of these residuals with respect to conditioning.

3.3 Experimental 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 by direct methods. 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 $B \in \mathbb{R}$ is formed by

$$B = L_0 L_0^T + \delta I_m, \qquad \delta > 0 \tag{3.6}$$

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 $\mathbf{x}$

$$\mathbf{x}^T(L_0 L_0^T)\mathbf{x} = (L_0^T\mathbf{x})^T(L_0^T\mathbf{x}) = \|L_0^T\mathbf{x}\|^2 \geq \mathbf{0} \tag{3.7}$$

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 (1.2.6), that the condition number of B when B is symmetric, is defined as:

$$\kappa(B) = \frac{\lambda_{\max}(B)}{\lambda_{\min}(B)} \tag{3.8}$$

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~~delta~~ 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 $\delta = 10^{-2}$, which gives a condition number of $\kappa(B) = 5.39 \times 10^5$.

Since B is symmetric and positive definite, we can compute its~~it's~~ Cholesky factorization $B = LL^T$ and construct A using a congruence transformation

$$A = LCL^T \tag{3.9}$$

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 3.3.1. *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 (3.9) holds, then the generalized eigenvalues $\Lambda(A, B)$ is similar to D*

Proof. Given the generalized problem

$$A\mathbf{v} = \lambda B\mathbf{v}, \qquad \mathbf{v} \neq \mathbf{0} \tag{3.10}$$

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 $\Lambda(A, B)$ is similar to D. □

The pseudocode for generating A and B is given as follows:[7]

[8] With the problem setup completed, and the algorithm described, in the next chapter,

[7] **[49]:** The algorithm environment floats, so you can't know for sure it will end up after "as follows." You should just put a reference like Algorithm \ref{alg:problem setup}. I don't know if that works with spaces in labels or not. I usually use underscores.

[8] **[50]:** I would just write $B = (L_0 L_0^T) + \delta I$ in the pseudocode

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))
  $B = (L_0 L_0^T) + \delta \cdot eye(m)$
  L = `cholesky`(B)
  $A = LCL^T$
  return (A, B)
end function

we shall discuss the results obtained in these experiments. [9]

[9] **[51]:** Overall, I like the pseudocode in this section.

CHAPTER 4
EXPERIMENTAL RESULTS AND DISCUSSION

In this chapter, we shall discuss the results obtained from the implementation of the algorithm for the

CHAPTER 5
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 methods, 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.

5.1 Summary of Key Findings

The experiments in this thesis have uncovered some interesting results regarding the spectral transformation lanczos algorithm for dense generalized eigenvalue problems. First, we have established that the generalized residuals increases for eigenvalues farther away from the shift, if the shift is not too large in magnitude ,validating the analytical error bounds proven for direct methods as observed in Michael Stewart 2024.

Secondly, our analysis of the eigenvalue sensitivity revealed the relationship between the conditioning of the matrices, the choice of shift parameter, 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 is 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 $A - \sigma B$"?

Thirdly, for the LU decomposition of $A-\sigma B$, we observe that the lanczos procedure was not stable and hence a significant amount of Ritz pairs did not converge, even with a low tolerance. This behavior is largely dependent on the conditioning of A and B. However, our results indicated that, the generalized residuals were insensitive to the conditioning of the problem.

5.2 Importance and Implications

The significance of this research can largely be categorized into 2:Theoretical advancements and practical applications.

5.2.1 Theoretical Contributions

From a theoretical perspective, this work advances our knowledge of spectral transformation, matrix conditioning and eigenvalue sensitivity in the context of dense generalized eigenvalue problems. Our results showed that the conditional bounds for direct methods, holds 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 trasnsformations affect the convergence of properties of Krylov subspace methods.

5.2.2 Practical Implications

Appendices

A Something

This is the appendix!

B Something Else

Another appendix!

REFERENCES

Jones, J., White, R. J., Quinn, S., Ireland, M., Boyajian, T., Schaefer, G., & Baines, E. K. 2016, ApJ, 822, L3