

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

by

Ayobami Adebesein

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, ill-conditioned. The Lanczos algorithm is a well-known iterative algorithm for computing the eigenvalues of a symmetric matrix and it works well 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  $\sigma$ . 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.

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 algorithm, ritz values, krylov subspaces, spectral transformation, orthogonality

Copyright by  
Ayobami Adebesein  
2025

Spectral Transformation Lanczos Algorithm for Symmetric-Definite Generalized Eigenvalue

Problems: A Comparative Analysis with Conditioning Insights

by

Ayobami Adebesein

Committee Chair:

Michael Stewart

Committee:

Jeter Russell

Vladimir Bondarenko

Alvin Das

Albert Einstein

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.

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 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 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 - 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. Adebessin, my siblings and several father and mother figures in my life. Mr. and Mrs. Olawuyi, Mrs. Abioye, Mr. Agboola for their unwavering support.

# TABLE OF CONTENTS

ACKNOWLEDGMENTS . . . . .	v
LIST OF TABLES . . . . .	ix
LIST OF FIGURES . . . . .	x
1 INTRODUCTION . . . . .	1
1.1 Background . . . . .	1
1.2 Mathematical Preliminaries . . . . .	2
1.2.1 Notation . . . . .	3
1.2.2 Floating Point Arithmetic . . . . .	3
1.2.3 Vector Norms . . . . .	4
1.2.4 General Matrix Norms . . . . .	5
1.2.5 Induced Matrix Norms . . . . .	6
1.2.6 Conditioning and Stability . . . . .	7
1.2.7 Congruence Transformation . . . . .	8
1.2.8 The Standard Eigenvalue Problem . . . . .	8
1.2.9 The Generalized Eigenvalue Problem . . . . .	10
1.2.10 Lanczos Algorithm . . . . .	11
1.2.11 Spectral Transformation . . . . .	12
1.3 Problem Discussion . . . . .	14
1.4 Motivation of Study . . . . .	15
1.5 Significance of Study . . . . .	15
2 LITERATURE REVIEW . . . . .	17
3 METHODOLOGY . . . . .	21
3.0.1 Spectral Transformation . . . . .	21
3.0.2 Lanczos decomposition . . . . .	22



4	RESULTS . . . . .	26
5	CONCLUSION . . . . .	27
	Appendices . . . . .	28
	A   Something . . . . .	29
	B   Something Else . . . . .	30
	REFERENCES . . . . .	31

**LIST OF TABLES**

Table 2.1	Insert Table Caption here . . . . .	20
-----------	-------------------------------------	----

## LIST OF FIGURES

Figure 2.1	An individual figure! . . . . .	18
Figure 2.2	Multiple figures! . . . . .	19

# 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 (GEP). The GEP takes the form  $Av = \lambda Bv$  where  $A$  and  $B$  are square matrices,  $\lambda$  is a generalized eigenvalue, and  $v$  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.

The GEP presents unique challenges as compared to the standard eigenvalue problem

## **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.

### 1.2.1 Notation

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

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

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

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

$\mathbf{x}, \mathbf{q} \in \mathbb{C}^m$  : denotes column vectors

Greek letters  $\alpha, \beta \dots$  : denotes scalars in  $\mathbb{C}$

$A^T$  : 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  $k$ th 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

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

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  $x \in \mathbf{F}$  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  $\mathbf{x}$ . 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  $\mathbf{F}$ , 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.

### 1.2.3 Vector Norms

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  $\mathbf{x}$  and  $\mathbf{y}$

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:

$$\|\mathbf{x}\|_1 = \sum_{i=1}^m |x_i| = 1,$$

$$\|\mathbf{x}\|_2 = \left( \sum_{i=1}^m |x_i|^2 \right)^{1/2},$$

$$\|\mathbf{x}\|_\infty = \max_{1 \leq i \leq m} |x_i|,$$

$$\|\mathbf{x}\|_p = \left( \sum_{i=1}^m |x_i|^p \right)^{1/p}, \quad (1 \leq p < \infty)$$

#### **1.2.4 General Matrix Norms**

Similar to a vector norm, a matrix norm is a function  $\|\cdot\| : \mathbb{C}^{m \times n} \rightarrow \mathbb{R}$  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

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

Let  $a_j$  be the  $j$ th column of  $A$ , equation 1.1 can be written as

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

In a more compact form ,we can rewrite it as

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

where  $\text{tr}(A)$  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:

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

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

The following matrix norms are useful:

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

- $\infty$ -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  $Q$ ,

$$\|QA\|_2 = \|A\|_2, \quad \|QA\|_F = \|A\|_F \quad (1.5)$$

### 1.2.6 Conditioning and Stability

Given any mathematical problem  $f : X \rightarrow Y$ , 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 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:

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

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" 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

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

where  $\sigma_1$  and  $\sigma_m$  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 ( $A \sim B$ ) if there exists an invertible matrix  $P$  such that

$$B = P^T A P \quad (1.8)$$

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

### **1.2.8 The Standard Eigenvalue Problem**

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,

$$A\mathbf{v} = \lambda\mathbf{v}, \quad \mathbf{v} \neq 0 \quad (1.9)$$

The (multi) set of all eigenvalues of  $A$  is called the *spectrum* of  $A$  and is denoted by  $\text{spec}(A)$ .

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,  $E_\lambda$  of  $A$  corresponding to an eigenvalue  $\lambda \in \text{spec}(A)$  as the set of all eigenvectors, together with the zero vector, associated with  $\lambda$  as follows:

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

The dimension of this vector space is called the *geometric multiplicity* of  $\lambda \in \text{spec}(A)$ . One of the many ways we can compute the eigenvalues of a matrix is by solving the characteristic 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 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  $A, B \in \mathbb{C}^{m \times m}$ , be any general square matrices. Then the set of all matrices  $A - \lambda B$  with  $\lambda \in \mathbb{C}$  is called a *pencil*. The *generalized eigenvalues* of  $A - \lambda B$  are the elements of the set  $\Lambda(A, B)$  defined by

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

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

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

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

$$A\mathbf{v} = \lambda B\mathbf{v} \quad (1.13)$$

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

$$B^{-1}A\mathbf{v} = \lambda\mathbf{v} \quad (1.14)$$

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 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.

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

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

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

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. In addition to that, we shall assume that  $A$  and  $B$  are dense matrices.

### **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 *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 or infeasible. It works by finding the "most useful" eigenvalues of the matrix - typically those at the extreme of the spectrum, and their eigenvectors. At its 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  $A \in \mathbb{C}^{m \times m}$ , and an initial vector  $v_1$ , the Lanczos algorithm produces a sequence of vectors  $v_1, v_2, \dots, v_n$  (where  $n$  is the number of iterations) that forms an orthonormal basis for the  $n$ -dimensional Krylov subspace

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

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 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 eigenvalues and eigenvectors are fundamentally tied to matrix operations. By applying a transformation to the matrix  $A$ , we can manipulate its eigenvalues and thus control which part of the spectrum, we are interested in. There are various types of spectral transformations, but the one that is 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  $\sigma$ . 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  $A \in \mathbb{R}^{m \times m}$ . 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 can pick a shift  $\sigma \in \mathbb{R}$  that is not an eigenvalue of  $A$ . The shifted and inverted 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  $\{(\lambda_j - \sigma)^{-1}\}$ , where  $\{\lambda_j\}$  are the eigenvalues of  $A$ . This shifts the spectrum of  $A$ , making the eigenvalues near  $\sigma$  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 shift-invert formulation of the problem is given by

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

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

The formulation shifts the spectrum of the generalized eigenvalues  $\Lambda(A, B)$  towards  $\sigma$ . Suppose  $\sigma$  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  $\sigma$  to the extreme part of the new spectrum, and by using an iterative method like the Lanczos algorithm, it is possible that the algorithm will converge to these extreme eigenvalues in the new spectrum.



### 1.3 Problem Discussion

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:

$$A\mathbf{v} = \lambda B\mathbf{v}, \quad \mathbf{v} \neq 0 \quad (1.18)$$

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.

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  $\sigma$ . To have a deep understanding of how well this method performs on 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 sensitivity of the residuals to ill-conditioning.

## 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 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 than the current ones we have today.

## 1.5 Significance of Study

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.

## CHAPTER 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  $\lambda$  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, explicitly 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 simultaneously 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.

And as an example of citing things, I'm going to cite a brilliant paper - (Jones et al.



Figure 2.1 An individual figure!

2016). See bibliography.bib for doing references.





(a)



(b)



(c)



(d)

Figure 2.2 Multiple figures!

Table 2.1: Insert Table Caption here

Blah	Blah	Blah
Stuff	Things	etc.
...	...	...

## CHAPTER 3

### METHODOLOGY

#### *3.0.1 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:

$$\beta A\mathbf{v} = \alpha B\mathbf{v}, \quad \mathbf{v} \neq 0 \quad (3.1)$$

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

Problem (3.1) can be reformulated as

$$\beta A\mathbf{v} = \alpha B\mathbf{v}, \quad \mathbf{v} \neq 0 \quad (3.2)$$

We have replaced  $\lambda$  with  $\alpha/\beta$  for convenience so that the generalized eigenvalues will be of the form  $(\alpha, \beta)$ . If  $\beta = 0$ , then the generalized eigenvalues  $\Lambda(A, B)$  will be infinite.

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. 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 :

$$C_b^T (A - \sigma B)^{-1} C_b \mathbf{u} = \theta \mathbf{u}, \quad \mathbf{u} \neq \mathbf{0} \quad (3.3)$$

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

Equation (3.2) 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  $\sigma$ , together with their corresponding eigenvectors. This will be explored in the next section.

### ***3.0.2 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  $A \in \mathbb{R}^{m \times m}$ , the summary of the lanczos algorithm is given as follows:

---

**Algorithm 1** Lanczos Algorithm for a Symmetric Matrix

---

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

```

1: function LANCZOS( $A, n, tol$ )
2:   Choose an arbitrary vector  $b$  and set an initial vector  $q_1 = b/\|b\|_2$ 
3:   Set  $\beta_0 = 0$  and  $q_0 = 0$ 
4:   for  $j = 1, 2, \dots, n$  do
5:      $v = Aq_j$ 
6:      $\alpha_j = q_j^T v$ 
7:      $v = v - \beta_{j-1}q_{j-1} - \alpha_j q_j$ 
8:     Full reorthogonalization:  $v = v - \sum_{i \leq j} (q_i^T v) q_i$ 
9:      $\beta_j = \|v\|_2$ 
10:    if  $\beta_j < tol$  then
11:      restart or exit
12:    end if
13:     $q_{j+1} := v/\beta_j$ 
14:  end for
15: end function

```

---

After the completion of algorithm 1, the  $\alpha$ 's and  $\beta$ 's are used to construct the tridiagonal matrix  $T_n$  and the vectors  $q_j$ 's are stacked together to form an orthogonal matrix  $Q_n$  given by:

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

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

The decomposition is given by

$$AQ_n = Q_n T_n + \beta_n q_{n+1} e_n^T \quad (3.4)$$

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  $\theta_i, i = 1, 2, \dots, n$  (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 spectral form of Algorithm (1).

---

**Algorithm 2** Spectral Lanczos Algorithm for (3.3)

---

**Require:**  $A = A^T$ ,  $B = B^T$ , with  $B$  being positive definite or semidefinite

**Require:** number of iterations:  $n$ , tolerance:  $tol$

**Require:**  $\sigma \in \mathbb{R}$ : shift not close to a generalized eigenvalue

```
1: function SPECTRAL_LANCZOS( $A, B, C_b, n, \sigma, tol$ )
2:   Choose an arbitrary vector  $b$  and set an initial vector  $q_1 = b/\|b\|_2$ 
3:   Set  $\beta_0 = 0$  and  $q_0 = 0$ 
4:   Precompute the  $LU$  factorization of  $A - \sigma B$ :  $lu\_factor = LU(A - \sigma B)$ 
5:   Factor:  $B = C_b C_b^T$ 
6:   for  $j = 1, 2, \dots, n$  do
7:      $u = Aq_j$ 
8:      $\alpha_j = q_j^T v$ 
9:      $v = u - \beta_{j-1}q_{j-1} - \alpha_j q_j$ 
10:    Full reorthogonalization:  $v = v - \sum_{i \leq j} (q_i^T v) q_i$ 
11:     $\beta_j = \|v\|_2$ 
12:    if  $\beta_j < tol$  then
13:      restart or exit
14:    end if
15:     $q_{j+1} := v/\beta_j$ 
16:  end for
17: end function
```

---

## CHAPTER 4

### RESULTS

## CHAPTER 5

## CONCLUSION

## 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