Study on Eigen Values and Eigen Vectors of Matrices: An Iterative Approach

By

Md. Sah Alam
Roll No : 1251552

A thesis submitted for the partial fulfillment of the requirements for the degree of Master of Philosophy in Mathematics

Khulna University of Engineering & Technology
Khulna 9203, Bangladesh.

May, 2016

Declaration

This is to certify that the thesis work entitled **" Study on Eigen values and Eigen vectors of matrices: an iterative approach "** has been carried out by **Md. Sah Alam** in the Department of Mathematics, Khulna University of Engineering & Technology, Khulna, Bangladesh. The above thesis work or any part of this work has not been submitted anywhere for the award of any degree or diploma.

…… ……..

Signature of Supervisor Signature of Candidate

Approval

This is to certify that the thesis work submitted by **Md. Sah Alam** entitled **"Study on Eigen values and Eigen vectors of matrices: an iterative approach"** has been approved by the board of examiners for the partial fulfillment of the requirements for the degree of **Master of Philosophy** in the Department of **Mathematics**, Khulna University of Engineering & Technology, Khulna, Bangladesh in May, 2016.

BOARD OF EXAMINERS

1. Chairman (**Supervisor**)
……
Prof. Dr. A. R. M. Jalal Uddin Jamali
Department of Mathematics
KUET, Khulna-9203.

2.
……..
Member
Head
Prof. Dr. M. M. Touhid Hossain
Department of Mathematics
KUET, Khulna-9203.

3.
……
Member
Prof.
Department of Mathematics
KUET, Khulna-9203.

4.
……
Member
Dr.
Associate Professor
Department of Mathematics
KUET, Khulna-9203.

5. Member (**External**)
……
Prof. Dr…….
Department of Mathematics
------- University

Dedication

To

My parents

Md. Idris Ali & Anura Khatun

Whose pains helped me see the light of the Earth

Acknowledgement

I am highly grateful to the Almighty for granting me time and effort to undertake this thesis and complete it.

I would like to tender my heartfelt gratitude to my thesis supervisor Dr. A.R.M. Jalal Uddin Jamali, Professor Department of Mathematics, Khulna University of Engineering & Technology for his close supervision, inspiration and guidance throughout the research and preparation of this thesis.

I also convey my profound gratefulness to the authority of Khulna University of Engineering &Technology for enrolling me as a fellow in M. Phil. Course. The serene academic environment of the University enthused me much to the course of research, which I must remember.

I profoundly fell proud to mention Professor Dr. Md. Bazlar Rahman, Professor Dr. Mohammad Arif Hossain, Professor Dr. Abul Kalam Azad, Professor Dr. M. M. Touhid Hossain and all other faculties of the Department of Mathematics for their all out cooperation for my study and preparation of the thesis paper. I cannot also forget the administration of the University for their all out support to bring my work to a desired end.

It will be gross ungratefulness on my part to avoid the sacrifices that my sister Lipika made throughout the period that this work required. My anti Saleha and cousin Hasan were also deprived of paternal affection many a time. I own my gratefulness to all of them.

Finally, I deeply remember the Director, Department of Military Lands & Cantonments, Ministry of Defense, my appointing authority for his kind permission to proceed for my higher study.

Abstract

There are some smart methods, available in the literature, which are able to find out all Eigen values. But those methods could not find corresponding Eigen vectors simultaneously. Power method and Inverse Power method are able to find out both Eigen-pairs simultaneously. Power method frequently used for finding only largest Eigen value and corresponding Eigen vector. On the other hand Inverse Power method is applied to find out only smallest Eigen value (or desire Eigen value) and corresponding Eigen vector. But Inverse Power method is computationally costly and some time it is unstable for the presence of inverse of the matrix. It is theoretically observed that if all Eigen values are either positive or negative, then without implement of Inverse Power method the modified (using shifting property) Power method is also able to find out smallest Eigen-pair. Here we have proposed Modified Hybrid Iterative Algorithm based on both Power method and Inverse Power method respectively to find out both largest and smallest Eigen-pairs simultaneously. Moreover several lemma regarding the proposed algorithm have been proposed. The proof of each lemma has also been given along with some suitable illustration. Several experiments have been performed to investigate the robustness and effectiveness of the algorithm as well as the lemma. The proposed algorithm is able to find out both (largest and smallest) Eigen-pairs successfully as well as efficiently.

Conference Proceedings

The following articles have been extracted from this thesis work:

1. A. R. M. Jalal Uddin Jamali and **Md. Sah Alam** "Largest as well as Smallest Eigen Pairs by Modified Power Method", 19th International Conference on Mathematics, Brac University, Dhaka, Bangladesh. December 18-20, 2015.

2. A. R. M. Jalal Uddin Jamali and **Md. Sah Alam** "Modified Iterative Algorithms for Eigen-pairs", Proceedings of the 1st International Conference on Mathematics and Its Applications (ICMA - 2015), Khulna University (KU), Khulna, Bangladesh. December 23, 2015.

Contents

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction

1.1 Field of Applications

Eigen values and vectors of a system are extremely important in physics and engineering where it is equivalent to matrix diagonalization. The theory and computation of Eigen value problems are among the most successful and widely used tools of applied mathematics and scientific computing and the ability to approximate these quantities numerically is important in a wide variety of applications. Matrix Eigen value problems arise naturally from a wide variety of scientific and engineering applications such common applications including structural dynamics, quantum chemistry, quantum mechanics, electrical networks, control theory and design, material science, the vibrations of membranes, in the separation of variables for the problems of heat conduction or acoustics or in the hydrodynamic stability analysis. Acoustics, earthquake engineering, Markov chains, pattern recognition, graph theory, stability analysis, the dynamics of elastic bodies, the physics of rotating bodies, small oscillations of vibrating systems, system identification, seismic tomography, principal component analysis, exploration and remote sensing, antenna array processing, geophysics, molecular spectroscopy, particle physics, structure analysis, circuit theory, Hopfield neural networks, mechanical system simulation and many other areas. For a partial list of these applications, see [Saad (1992), Trefethen and Embree (2005), Xue (2009), Chu and Golub (2002), Joseph (1992), Hald (1972), Parker and Whaler (1991), Li (1997), Elhay and Ram (2002), Andrew (194), Ying and Li (2012), Koledin and Stani (2013) , bioinformatics (Kato and Nagano (2010)), Vert et al. (2007), image recognition Chopra et al. (2005), Guillaumin et al. (2009) and Hoi et al. (2006)].

An inverse Eigen value problem is to determine a structured matrix from a given spectral data. Inverse Eigen value problems arise in many applications, including control design, system identification, seismic tomography, principal component analysis, exploration and

remote sensing, antenna array processing, geophysics, molecular spectroscopy, particle physics, structure analysis, circuit theory, Hop-field neural networks, mechanical system simulation and so on [Jian (2004), Chu and Golub (2002)].

Nonlinear Eigen value problems arise in a multitude of diverse applications from science and technology, such as acoustic field simulations [Mehrmann and Schroder (2011)], computational quantum chemistry [Voss (2006)], structural dynamics [Singh and Ram (2002)], electromagnetic modeling of particle accelerators [Liao (2010)], vibrations of fluid-solid structures [Voss (2003) and Unge (2013)] or stability analysis of time-delay systems [Michiels and S.-I.Niculescu (2007) , Jarlebring (2008)]. For a more comprehensive overview of sources for nonlinear Eigen value problems, see [Mehrmann and Voss (2004)].

The spectral properties of a physical system govern its dynamical performance. Hence the computation of Eigen values enables the basic understanding of the underlying physical system. Stochastic matrices, which are a type of special nonnegative matrices are with each row sum to 1. Nonnegative stochastic matrices are widely used in game theory, Markov chains and theory of probability, probabilistic algorithms, discrete distributions, categorical data, group theory, matrix scaling and economics.

It is important to note that various areas of science and engineering seek multiple pairs for reasons other than algorithmic gains. In quantum mechanics Eigen values are possible measurement results of an observable represented by an operator. In nuclear engineering, a dominance ratio distinct from unity is an acceptance qualifier for various nuclear criticality safety assessments and nuclear reactor designs [Spanier and Carlo (1969)]. In statistical physics, a dominance ratio nearing unity, on the other hand, is often a condition sought. Near a continuous phase transition, $\lambda_2 \rightarrow \lambda_1$ and λ_2/λ_1 controls the microscopic spatial correlations among physical degrees of freedom 2 [Thompson (1972)]. Today, an important topic in quantum statistical mechanics is quantum critical phenomena, phase transitions driven by zero-point motion at zero temperature [Sachdev (1999)]. Here, it is the two smallest Eigen values of the Hamiltonian matrix describing the physical system that are of interest. The

quantum critical phenomenon construct, while supplemented by a few exact solutions to some very simple problems, is largely phenomenological in part because of the inability to compute λ_2 for models of direct physical relevance.

Eigen values were used by Claude Shannon to determine the theoretical limit to how much information can be transmitted through a communication medium like your telephone line or through the air. This is done by calculating the Eigen vectors and Eigen values of the communication channel (expressed a matrix). The Eigen values are then, in essence, the gains of the fundamental modes of the channel, which themselves are captured by the vectors.

Eigen vectors are fundamental to principal components analysis which is commonly used for dimensionality reduction in face recognition and other machine learning applications. Eigen vectors can also be used for latent semantic analysis, a NLP technique for extracting topics and concepts from text documents.

In control theory, the Eigen values of the system matrix of a linear system tell you information about the stability and response of your system. For a continuous system, the system is stable if all Eigen values have negative real part (located in the left half complex plane). For a discrete system, the system is stable if all Eigen values have magnitude less than 1 (inside the unit circle in the complex plane).

The Eigen values and vectors of a matrix are often used in the analysis of financial data and are integral in extracting useful information from the raw data. They can be used for predicting stock prices and analyzing correlations between various stocks, corresponding to different companies. They can be used for analyzing risks. There is a branch of Mathematics, known as Random Matrix Theory, which deals with properties of Eigen values and Eigen vectors, that has extensive applications in Finance, Risk Management, Meteorological studies, Nuclear Physics, etc.

In statistics, the Eigen vectors of your data set matrix correspond to directions of maximum variance, ordered in decreasing marginal increase in variance by decreasing corresponding

Eigen values. This is the main idea behind principal component analysis (PCA), a dimensionality reduction trick often used in machine learning and AI (artificial intelligence).

The Eigen values are used to determine the natural frequencies (or frequencies) of vibration, and the Eigen vectors determine the shapes of these vibration modes. Most structures from buildings to bridges have a natural frequency of vibration. It means all these structures have their own system of vibrations and frequencies. Now external forces like wind and earthquake may cause these structures to undergo vibrations. In case the frequency of these vibrations becomes equal to the natural frequencies of these structures, vibrations with large amplitudes are set up. It is a phenomenon called Resonance. This can lead to the collapse of the structure by a process called aero elastic flutter. One very famous example of the collapse of a structure due to these phenomena is the Tacoma Narrows Bridge (1940) in which the wind provided an external periodic frequency that matched the bridge's natural structural frequency. So vibration analysis of these structures is done at the time of their design using Eigen values and Eigen vectors.

The Eigen values can also be used to determine if a structure has deformed under the application of a particular force. Eigen values for the structure are measured before and after the application of force. If a change in the Eigen values is observed, it means the structure has undergone deformation. This is just one of the fields that make practical use of the Eigen values of a matrix.

A recent application is the search engine Google (2005), which uses the vector corresponding to the Eigen value one for an extremely large sparse stochastic matrix. Google uses the vector corresponding to the maximal Eigen value of the Google matrix to determine the rank of a page for search.

The increasing number of applications and the ever-growing scale of the problems have motivated fundamental progress in the numerical solution of Eigen value problems in the past few decades. New insights and extensions of existing computational methods usually go hand in hand with the development of new algorithms and software packages.

1.2 Eigen Value Problems

Eigen values are a special set of scalars associated with a linear system of equations (i.e., a matrix equation) that are sometimes also known as characteristic roots, characteristic Eigen values, proper Eigen values, or latent roots [Hoffman and Kunze (1971), Marcus and Minc (1997)]. The determination of the each Eigen value is paired with a corresponding Eigenvector. The Eigen value and corresponding Eigen vector are denoted as Eigen pair. The decomposition of a square matrix **A** into Eigen values and vectors is known in this work as decomposition, and the fact that this decomposition is always possible as long as the matrix consisting of the vectors of **A** is square is known as the decomposition theorem.

Before define Eigen value and Eigen vector we would like to define linear transformation. According to the Linear algebra, let **T** is a linear operator on a vector space **V** over a field F and if there exist Eigen vector **v**,**w** ϵ **V** such that

T(**v**) = **w**

Then this equation form a linear transformation where **w** maps to **v** by the linear operator **T.** In many contexts, a vector can be assumed to be a list of real numbers (called coordinates), written vertically with brackets around the entire list, such as the vectors $\mathbf{v} = [v_1, v_2, \cdots, v_n]^{/}$ and $\mathbf{w} = [w_1, w_2, \cdots, w_n]^{/}$. Now if operator **T** be a square matrix of order *n* say **A**, then we have **Av** = **w,** i.e.

$$\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_3 \end{bmatrix} = \begin{bmatrix} A_{1,1} & A_{1,2} & \dots & A_{1,n} \\ A_{2,1} & A_{2,2} & \dots & A_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n,1} & A_{n,2} & \dots & A_{n,n} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

where for each index *i*,

$$w_i = A_{i,1}\, v_1 + A_{i,2}\, v_2 + \cdots + A_{i,n}\, v_n = \sum_{j=1}^{n} A_{i,j}\, v_j$$

In general, if $\mathbf{v}_j$ are not all zeros, the vectors **v** and **Av** i.e. **w** will not be parallel. Almost all vectors change direction, when they are multiplied by **A**. Certain exceptional vectors **v** are in the same direction as **Av**. Those are the "vectors" Multiply an vector by **A** and the vector **Av** is a number of times the original **v**. i.e. when they are parallel (that is, when there is some real

number λ such that) $\mathbf{A\upsilon} = \lambda\mathbf{\upsilon}$ we say that $\mathbf{\upsilon}$ is an vector of $\mathbf{A}$. In that case, the scale factor λ is said to be the Eigen value corresponding to that vector.

The matrix **A** may change both the direction and the magnitude of an arrow **v** in three-dimensional space. However, if **v** is a vector of **A** with Eigen value λ, the operation may only change its length and either keep its direction or flip it (make the arrow point in the exact opposite direction). The Eigen value tells whether the special vector **v** is stretched or shrunk or reversed or left unchanged when it is multiplied by **A**. Specifically, the length of the arrow will increase if $|\lambda| > 1$ remain the same if $|\lambda| = 1$ and decrease it if $|\lambda| < 1$. Moreover, the direction will be precisely the same if $\lambda > 1$ and flipped if $\lambda < 1$. If $\lambda = 1$, then the length of the arrow becomes zero.

Now if **v** be the vector in two dimensional spaces then geometry of the linear transformation is illustrated by the following example. For the transformation matrix

$$\mathbf{A} = \begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix},$$

The vector $\mathbf{\upsilon} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$, is an vector with Eigen value 2. Indeed,

$$\mathbf{A\upsilon} = \begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 3.1 + 1.(-1) \\ 1.1 + 3.(-1) \end{bmatrix} = \begin{bmatrix} 2 \\ -2 \end{bmatrix} = 2 \begin{bmatrix} 1 \\ -1 \end{bmatrix}.$$

On the other hand the vector $\mathbf{u} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ is not a vector, since

$$\begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 3.0 + 1.1 \\ 1.0 + 3.1 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \end{bmatrix},$$

and this vector is not a multiple of the original vector **u**.

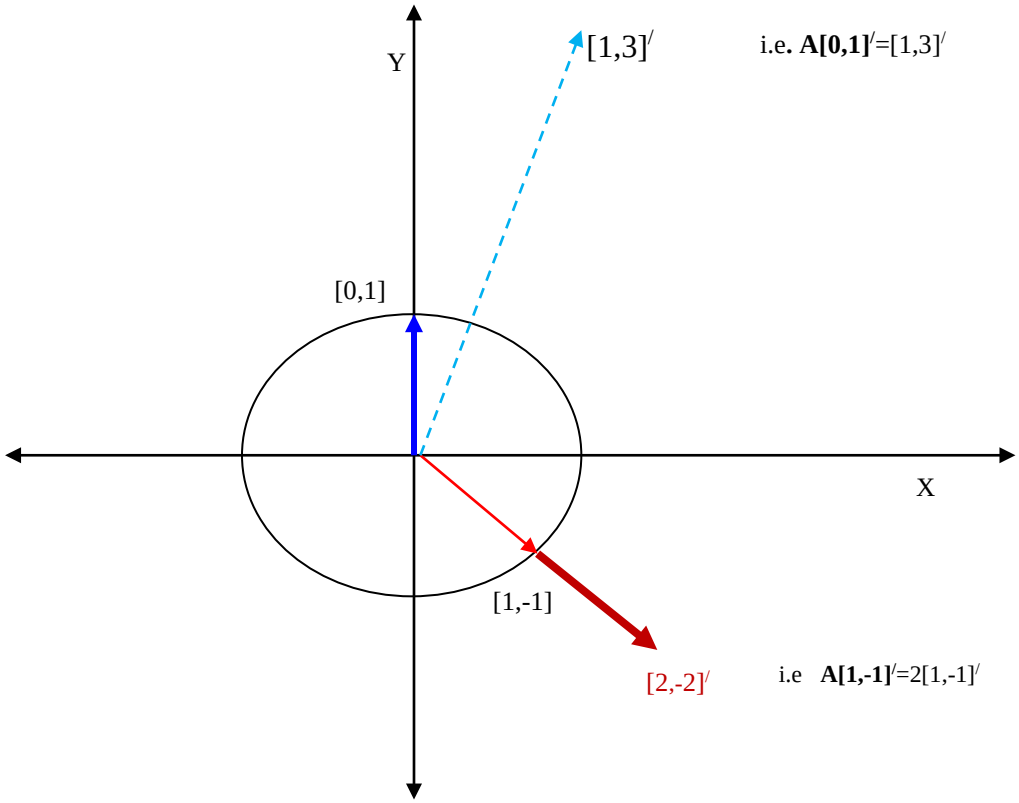


Figure 1.1: Geometry of the linear transformation and Eigen value problem

Now formally we will define Eigen value and Eigen vector. According to the Linear algebra, the definition of Eigen value problem is defined as follow: Let **T** is a linear operator on a vector space **V** over a field F. If there exist a scalar $\lambda \in F$ a non zero vector $\mathbf{v} \in \mathbf{V}$ such that

$$\mathbf{T}(\mathbf{v}) = \lambda\mathbf{v} \qquad (1.1)$$

Then the equation (1.1) forms an Eigen value problem. Here λ is called an Eigen value and such a vector **v** is called an Eigen vector of **T** corresponding to the Eigen value λ. The following have the same meaning: Characteristic root, proper root, latent root and spectral Eigen value. Similarly the following have the same meaning: Characteristic vector, proper vector, latent vector, Eigen vector.

Now if the operator **T** is a matrix **A** and let **A** is a square matrix of order *n* then we also write

$$\mathbf{Av} = \lambda\mathbf{v} \qquad (1.2)$$

Here **A** is matrix representation of linear transformation **T** in any basis. Now the matrix polynomial **A**-λ**I** of the first degree is called characteristic matrix of **A**, **I** being unit matrix of order *n*. Then determinant |**A**-λ**I**| is called characteristic polynomial of **A**. Clearly this determinant is ordinary polynomial of degree *n*. The equation |**A**-λ**I**|=0 is called characteristic equation of **A**. The roots of this equation are called characteristic roots or latent roots or Eigen values. The set of Eigen values of **A** is called the spectrum of **A**. The absolute Eigen value of largest Eigen value is called spectral radius. Again let **A** be a linear transformation represented by a matrix **A**. If there is a vector $\mathbf{x} \in \boldsymbol{R^n}$ such that

$$\mathbf{Ax} = \lambda\mathbf{x} \tag{1.3}$$

For some scalar λ, then λ is called the Eigen value of **A** with corresponding vector **x**. This is called Eigen value problem. Equation (1.3) can be written compactly as

$$(\mathbf{A} - \lambda\mathbf{I})\mathbf{x} = \mathbf{0} \tag{1.4}$$

where **I** is the identity matrix and **0** is null vector. According to the Cramer's rule, a linear system of equations has nontrivial solutions if and only if the determinant of (**A** – λ**I**) vanishes, so the solutions of equation (1.4) are given by

$$\det(\mathbf{A} - \lambda\mathbf{I}) = \mathbf{0} \tag{1.5}$$

This equation (1.5) is known as the characteristic equation of **A** and the left-hand side is known as the characteristic polynomial.

Also the Eigen value problems frequently arise in the engineering field where physical problems are model by Ordinary Differential Equation (ODE). Suppose we have an ODE system as follows

$$\frac{\boldsymbol{dx}}{dt} = \mathbf{Px} \tag{1.6}$$

where **P** is a constant square matrix and **x** be vector. By taking $\mathbf{X} = \mathbf{v}e^{\lambda t}$ as a trial solution in equation (1.6), we have

$$\lambda\boldsymbol{\nu} = \mathbf{P}\boldsymbol{\nu} \tag{1.7}$$

where λ be the Eigen value and **v** is the corresponding Eigen vector.

Several methods are available in literature to find out the Eigen values and Eigen vectors. Some of them are direct methods and some of them are iterative methods. Direct methods usually provided Eigen values only. For finding corresponding Eigen vectors, much more efforts are necessary. It is worthwhile to mention here that most of the methods find either

only Eigen values or only Eigen vectors. When the dimension of characteristics matrix is very high or the characteristic the matrix is sparse, iterative methods are, in general, computationally efficient.

1.3 Iterative Methods

In computational mathematics, an iterative method is a mathematical procedure that generates a sequence of improving approximate solutions for a class of problems. A specific implementation of an iterative method, including the termination criteria, is an algorithm of the iterative method. An iterative method is called convergent if the corresponding sequence converges for given initial approximations. A mathematically rigorous convergence analysis of an iterative method is usually performed; however, heuristic-based iterative methods are also common. In the problems of finding the root of an equation (or a solution of a system of equations), an iterative method uses an initial guess to generate successive approximations to a solution. In contrast, direct methods attempt to solve the problem by a finite sequence of operations. In the absence of rounding errors, direct methods would deliver an exact solution (like solving a linear system of equations $\mathbf{Ax} = \mathbf{b}$ by Gaussian elimination). Iterative methods are often the only choice for nonlinear equations. However, iterative methods are often useful even for linear problems involving a large number of variables (sometimes of the order of millions), where direct methods would be prohibitively expensive (and in some cases impossible) even with the best available computing power.

In mathematics, the power iteration is an Eigen value algorithms: given a matrix **A**, the algorithm will produce a number λ (the Eigen value) and a nonzero vector **v** (the vector), such that $\mathbf{Av} = \lambda\mathbf{v}$. The algorithm is also known as the Von Mises iteration [Marcus and Minc (1988)].The Power Iteration is a very simple algorithm. It does not compute matrix decomposition and hence it can be used when **A** is a very large sparse matrix. However, it will find only one Eigen value (the one with the greatest absolute Eigen value) and corresponding Eigen vectors. The convergence rate of the Power method depends on $[\lambda_2/\lambda_1]$, where λ_1 is the largest Eigen value whereas λ_2 is the second largest Eigen value of **A** in magnitude. This ratio

is generally smaller than 1, allowing adequate convergence. But there are cases where this ratio can be very close to 1, causing very slow convergence. For detailed discussions on the power method, see [Demmel (1997), Golub and Loan (1996), Parlett and Cliffs (1980)].

Although the Power iteration method approximates only one Eigen value of a matrix, it remains useful for certain. For instance, uses it to calculate the documents in their search engine, and uses it to show users recommendations of who to follow. For matrices that are well-conditioned and as sparse as the Web matrix, the Power iteration method can be more efficient than other methods of finding the dominant (largest) vector. The method can also be used to calculate the spectral radius of a matrix by computing the Rayleigh quotient. Though Power method is applicable in various field of application where only largest Eigen value pairs are important, but there are many fields in which both largest as well smallest Eigen value and corresponding Eigen vectors are very important. Moreover in some cases the nature of the signs of spectrum is valuable.

1.4 Goal of Thesis

At first we have studied rigorously regarding Eigen value problems and some important methods mainly iterative methods to earn adequate knowledge about the study. Our main goals have pointed out bellow:

(a) We have developed a Modified Hybrid Iterative Algorithm based on Power method and Inverse Power method to find out largest Eigen pair and smallest Eigen pair according to the absolute magnitude. Moreover the algorithm is able to find out second largest Eigen pair in some cases.

(b) Some experiments have been performed to verify the effectiveness of the proposed algorithm.

(c) Four Lemma also have been proposed relevant to the proposed algorithm. Moreover, these proofs of the Lemma are given along with some illustrations.

1.5 Structure of the Thesis

After **Chapter I** in which the introduction of the research works is presented, the literature review is discussed in **Chapter II**. **Chapter III** presents the review of the properties of Eigen value and Eigen vector and **s**ome important iterative methods**.** Also detail discussion of power method and inverse Power method are discussed in this chapter. Moreover the rate of convergence and Limitations of Power method are also briefly discussed in **Chapter III.** In **Chapter IV**, the proposed algorithm is presented and discussed a bit elaborately. Some experiments are performed to check the validity as well as effectiveness of the proposed algorithms. Some Lemma related to the algorithm is proposed in **Chapter V.** The proof of the Lemma are also presented in this chapter. The Lemmas is also implemented in the proposed algorithm with some illustrations. Finally concluding remarks and brief discussion about the research works are given in **Chapter VI.** The list of the references is presented at the end of the thesis as well.

CHAPTER 1I

Literature Review

2.1 Introduction

In view of the central importance of the Eigen value problem in so many field of pure and applied mathematics, much thought has been devoted to designing of efficient methods and effective properties by which the Eigen values as well as Eigen vectors of a linear operator may be found. That Linear operator may be of algebraic or continuous type; that is a matrix **A** differential operator or a kernel function.

2.2 Background

Numerical linear algebra is a very active field of research. Many problems are challenging of themselves and in addition, much of scientific computing depends critically in one way or another on numerical linear algebra algorithms. Not only do the more classical scientific computational models for physical or engineering problems depend on linear algebra kernels, but many modern applications, such as information retrieval and image restoration, profit from numerical linear algebra results. These factors have motivated numerical linear algebra research throughout the entire 20^{th} century.

The field has blossomed, especially since the introduction of the modern computer, roughly from the early 1950. This is evident from the large number of scientific journals in which articles in this area appear: SIAM on Matrix Analysis and Applications (SIMAX), Linear Algebra and its Applications (LAA), Numerical Linear Algebra with Applications (NLAA), are completely devoted to this specialty. Articles on numerical linear algebra, theoretical as well as applied, regularly appear in journals such as BIT, SIAM Numerical Analysis, SIAM J.

on Scientific Computing, J. on Computational and Applied Mathematics, J. Applied Numerical Mathematics, Numerical Mathematics, Numerical Algorithms, Mathematics of Computation, Parallel Computing, ACM Transactions on Mathematical Software, Computing, J. Inst. Math. Application., SIAM Review, IMA J. Num. Anal., and several others in more application oriented directions, such as J. Computational Physics and engineering journals. And from, for instance, the bibliography in Golub and Van Loan's Book (1989), one can see how many papers are referenced from these and other sources. A quick glance through the contents of the average 60 papers per year in SIMAX shows that roughly 40% of the papers are associated with Eigen value problem research and it is likely that this holds more or less for the many papers per year that focus on numerical linear algebra.

The paper of Gene et al. (2000) has sketched the main research developments in the area of computational methods for Eigen value problems during the 20th century. In this paper [Gene et al. (2000)] authors have considered the algorithmic developments from a historical point of view and to indicate how the recent powerful techniques are the result of many smaller steps. This will also help to show how many of the algorithms are interrelated. The reader who is interested in methods that have played a role but that are at present no longer considered to be on the main track, is referred to Wilkinson's books (1965). In addition, Parlett (1980) gives interesting historical information on older methods that still have some significance from a theoretical point of view.

In fact, this was already recognized by Jacobi who, in 1846, computed the Eigen values of symmetric matrices by rotating the matrix to a strongly diagonally dominant one. Note that Jacobi's techniques are still relevant and have led to popular and powerful algorithms. Another longstanding method that is of great significance and serves as the basis for many algorithms is the Power iteration. The method is based on the idea that if a given vector is repeatedly applied to a matrix, and is properly normalized, then ultimately, it will lie in the direction of the Eigen vector associated with the Eigen values which are largest in absolute value. The rate of convergence for the Power iteration depends on the ratio of the second largest Eigen value (in absolute value) to the largest Eigen value (in absolute value) and for

many applications this leads to unacceptably slow convergence. The method can be problematic if one wants to compute a number of External Eigen values. Anyway including Power method there exist some most frequently used iterative methods as well as direct methods such as Krylov methods, inverse Power iteration, QR-method etc.

2.3 Review of Methods

The problem (1.3) corresponds to finding the zeroes of the characteristic polynomial det (A − λI) = 0 of **A** of order *n*. It is well-known that for $n \geq 5$ there is no expression for the roots of this polynomial for a general **A**; therefore, determining the exact Eigen values is generally not possible. However, there are many numerical methods that give very good approximations to Eigenvectors and hence Eigen values of a given matrix. Eigen value problems of moderate size, which means that a full $n \times n$ matrix can be stored conveniently, are often solved by direct methods, by which we mean methods where similarity transformations are applied until the Eigen value estimates can be easily found.

The best known direct method which is able to find all Eigen values is the QR-Algorithm, based on the QR decomposition of a matrix [Bai et al. (1999) and Demmel (1997)] which is also implemented in the Matlab function **eig ( ).** The QR Iteration algorithm for computing the Eigen values of a general matrix came from an elegantly simple idea that was proposed by Heinz Rutishauser in 1958 and refined by Francis in 1961-1962. The QR algorithm approximates the whole spectrum and the number of iterations needed is of order O (n^3), where *n* is the size of the matrix, which becomes very large for large problems. There is another disadvantage of the QR method. If matrices are sparse, that is, the number of non-zero elements is small compared to the number of zero entries, and the matrix is structured, then the QR method generates matrices in which the sparse structure of the original matrix disappears. This leads to filled in and an increasing storage requirement as the algorithm proceeds.

In many applications it is not necessary to calculate the complete Eigen value decomposition of a matrix. Often only a few Eigen values are of interest. Moreover in some physical situation in where Eigen vectors along with Eigen values play important role. Consequence iterative methods come to lime light. By iterative methods we mean methods based on matrix vector multiplications using the original sparse matrix so that the sparse matrix storage and structure can be used to advantage. Hence, subspace algorithms are suitable for large sparse matrices. All subspace algorithms have the following structure in common:

1. Generate a sequence of subspaces $S_1, S_2, \ldots$
2. For each subspace S_i of dimension i construct a matrix $\mathbf{H}_i \in \mathbf{C}^{i \times i}$ which is the restriction and projection of $\mathbf{A}$ onto the subspace S_i.

The matrices $\mathbf{H}_i$ are usually constructed with the Rayleigh-Ritz procedure, which can be described as projecting and restricting the full matrix $\mathbf{A}$ onto the subspace. Then the Eigen values of the projected matrix are called Ritz values which are approximations to the wanted part of the spectrum. The corresponding Eigen vectors of $\mathbf{A}$ are called Ritz vectors and they represent approximations to the exact Eigen vectors of $\mathbf{A}$. Different subspace methods are distinguished from the way the subspaces are generated. The dimension of the subspace may fixed are variable. If the dimension of the subspace is fixed to one then the most common methods obtained are the Power method and Rayleigh quotient iteration (see for example [Parlett (1980) and Wilkinson (1965)] for details). The Power method can be extended to subspaces with higher, but fixed dimension, where it is called subspace or simultaneous iteration which can be seen as a block Power method. For details see [Van Loan (1996) and Saad (1992)]. A further class of subspace methods is the one that uses nested subspaces of increasing dimension. Usually one starts with a subspace of dimension one and increases this dimension by one at each iteration step. Among the most popular of these methods are the Lanczos method (1950) for symmetric matrices and the Arnoldi method [Lehoucq and Meerbergen (1999), Sorenson and Yang (1998)] for nonsymmetric matrices. These methods are Krylov subspace methods. More details on Arnoldi and Lanczos methods can be found in Golub and Van Loan (1996), Saad (1992), Bai et al. (1999) and Trefethen and Bau (1997). The methods of Lanczos and Arnoldi have lead to the development of many other algorithms. For example, both methods can be generalized to block Lanczos and block Arnoldi

algorithms, by working with p-dimensional subspaces instead of vectors. The iteration starts with a p-dimensional subspace and the dimension is increased by p at each step.

There exist further subspace algorithms with increasing subspace dimension, where the subspace is expanded without using Krylov subspaces. A Newton iteration step or an approximate Newton iteration step can be applied to obtain a new direction. Examples for this approach are the Davidson method and the Jacobi-Davidson method, see [Sleijpen and Vorst (1996) and Hochstenbach and Notay (2004)] for details.

Anyway a well known iterative method for finding both Eigen value and corresponding Eigen vectors is called Power iteration or simply Power method [Marcus and Minc (1988)]. For our discussion of the Power method, we have borrowed material from Householder's book [Householder (1964)]. The Power method, for general square matrices, is the simplest of all the methods for solving for Eigen values and Eigen vectors. The basic idea is to multiply the matrix $\mathbf{A}$ repeatedly by a well-chosen starting vector, so that the component of that vector in the direction of the Eigen vector with largest Eigen value in absolute value is magnified relative to the other components. Householder called this Simple Iteration and attributed the first treatment of it to Muntz in 1913. Bai (1995) attributes the Power method to Von Mises (1929) and acknowledges Muntz for computing approximate Eigen values from quotients of minors of the explicitly computed matrix $\mathbf{A}_k$, for increasing values of k. For a careful analytic treatment of the Power method, Householder acknowledged work by Ostrowski and Werner Gautschi; the reader can find a fairly complete treatment in Wilkinson's book (1965) together with the proper references.

A compact description and analysis of these ideation techniques was given by Parlett (1980). The Power method and the Inverse Power method, in their pure form are no longer competitive methods even for the computation of a few Eigen pairs, but they are still of interest since they are explicitly or implicitly part of most modern methods such as the QR method and the methods of Lanczos and Arnoldi. These methods evolved in some way or another from the Power method and some of the techniques that were suggested as

improvements to the Power method are still in use as acceleration techniques for modern iterative methods. One of these idea is to work with polynomials of **A**, with the purpose of damping unwanted parts of the spectrum. Detailed discussions on the power method can be found in Demmel (1997), Golub and Van Loan (1996), Parlett (1980), SIAM Philadelphia (1997).

Although the power iteration method approximates only one Eigen value of a matrix, it remains useful for certain. For instance, uses it to calculate the documents in their search engine and uses it to show users recommendations of who to follow. For matrices that are well-conditioned and as sparse as the Web matrix, the power iteration method can be more efficient than other methods of finding the dominant (largest) Eigenvector. Some of the more advanced Eigen value algorithms can be understood as variations of the power iteration. For instance, the inverse iteration method [Demmel and James (1997)] applies power iteration to the matrix. The method can also be used to calculate the spectral radius of a matrix by computing the Rayleigh quotient. For detailed discussions on the Power method, see [Demmel (1997), Golub and Loan (1996), Parlett and Cliffs (1980)].

Another possibility is working with properly updated shifts μ in the inverse process and in particular, if one takes the Rayleigh quotient with the most recent vector as a shift, then one obtains the Rayleigh quotient iteration. According to Parlett (1980), Lord Rayleigh used in the 1870 a less powerful technique. He did a single shift-and-invert step with a Rayleigh quotient for an Eigen vector approximation but with a unit vector as the right-hand side. (This saves the refactoring of the matrix ($\mathbf{A} - \mu\mathbf{I}$) at each iteration). The modern RQI in which one takes the most current Eigen vector approximation as the right-hand side, leads to very fast convergence. Ostrowski (1958) studied the convergence properties for variance of RQI for the symmetric and unsymmetrical case. He was able to establish cubic convergence in both cases under various circumstances (in the unsymmetrical case for a properly generalized Rayleigh quotient). These results are essential for the understanding of modern iterative techniques that are based on (approximate) shift-and-invert strategies (for example, the Jacobi-Davidson method).

Since orthogonal reduction techniques often evidence superior stability properties, it became apparent that the LR factorization should be replaced by a QR factorization. This leads to one of the most popular and powerful methods of our time for Eigen value problems: The QR method for computing all of the Eigen values and associated Eigen vectors of a dense symmetric matrix. (In fact, the QR method has essential enhancements that make the method really powerful). With the number of vectors less than n, this Power method in combination with QR orthogonalization is known as the Simultaneous Iteration method; Rutishauser (1969), studied this method in 1969. Its convergence behavior for general unsymmetrical matrices was studied by Stewart (1976) in 1976. Stewart also developed a subroutine, based on simultaneous iteration, for the computation of a dominant invariant subspace.

The collection of vectors generated by the Power method defines Krylov subspaces of increasing dimension. This motivated Krylov to try to determine the characteristic polynomial of a matrix by inspecting the dependence of a full set of these vectors. This procedure may fail because the system of equations is highly ill-conditioned but this can be repaired by orthogonal zing each new vector to the previous vectors and applying **A** onto the last constructed vector. This iteration process is known as the Lanczos method for symmetric matrices and Arnoldi's method for unsymmetrical matrices.

Parlett (1980) however, described situations where Simultaneous Iteration is still competitive. For instance, if we can store only a limited number of n-vectors in fast memory or if the relative gap between the desired Eigen values and the others is great than Simultaneous Iteration is very useful.

Booth (2003, 2006) were developed Monte Carlo simulations of steady state neutron transport in nuclear reactors. Initially, he proposed a novel modification of the power method that has produced up to 10 Eigen pairs for simple test problems.

Karoui (2006) has made very important progress in recent years in the understanding of the behavior of the Eigen values of a large number of large-dimensional random matrices. He found many new results concern the fluctuation of these Eigen values. These new findings show great promise for applications, in particular, in Statistics. In this paper, his focus is on fine convergence properties of the largest Eigen value of a class of random covariance matrices.

In this paper, McCormick and Noe (1977) developed an accelerated version of simultaneous iteration for partial solution of the Eigen problem. Their proposed algorithm convergence together with sharp error bounds is obtained. They were able to obtain some important theoretical results regarding symmetric Eigen problem where the algorithms are shown to be improvements over existing techniques. In application to a given matrix, the accelerated convergence can often be dramatic, particularly when a clustering of the dominant Eigen values occurs. Simultaneous iteration and its variants are natural extensions of the power method, which on the other hand is plagued by slow convergence in the presence of clustering. These techniques [McCormick and Noe (1977)] attempt to capitalize on the assets of the power method while at the same time overcoming its major handicap.

Panju (2011) examined some numerical iterative methods for computing the Eigen values and Eigen vectors of real matrices. The five methods examined here range from the simple power iteration method to the more complicated QR iteration method. The derivations, procedure, and advantages of each method are briefly discussed in Panju (2011).

Computing Eigen pairs of large matrices is a ubiquitous problem in computational physics. Author, in the paper [Gubernatis and Booth (2008)], presented several refinements of the basic Power method that enable the efficient and accurate computation of multiple extremely Eigen values of very large matrices. Ultimately, their objective is producing Monte Carlo versions of such methods for matrices whose orders are so large that even the Eigen vectors be stored in computer memory is very hard. For such problems, the computation of a basic vector quantity as the inner product is generally either very inefficient or impractical. Also

Gubernatis, Booth (2008) proposed several refinements of the Power method that enable the computation of multiple external Eigen pairs of very large matrices by using Monte Carlo simulation method.

Gotze et al. (1993) proposed an approximation to the Exact Jacobi method and claimed that it is more efficient. Besides the Jacobi-based method, the Algebraic Method [Waerden and Van der (1991)] offers an alternative approach that may be more efficient for some restricted classes of Eigen value problems.

Two-norm normalized inverse, shifted inverse, and Rayleigh quotient iteration are well-known algorithms for approximating an Eigenvector of a symmetric matrix. Tapia et al. (2015) have established rigorously that each one of these three algorithms can be viewed as a standard form of Newton's method from the nonlinear programming literature, followed by the normalization. This equivalence adds considerable understanding to the formal structure of inverse, shifted inverse, and Rayleigh quotient iteration and provides an explanation for their good behavior despite the possible need to solve systems with nearly singular coefficient matrices; the algorithms have what can be viewed as removable singularities. Moreover Tapia et al. (2015), presented a thorough historical development of these Eigen value algorithms. Utilizing their equivalences, authors constructed traditional Newton's method theory analysis in order to gain understanding as to why, as normalized Newton's method, inverse iteration and shifted inverse iteration are only linearly convergent and not quadratically convergent, and why a new linear system need not be solved ateach iteration. They also investigate why Rayleigh quotient iteration is cubically convergent and not just quadratically convergent.

2.4 Some important fields of applications

An Inverse Eigen value Problem (IEP) is to construct a matrix which possesses both proscribed Eigen values and desired structure. Inverse Eigen value problems arise in broad application areas such as control design, system identification, principle component analysis, structure analysis etc. There are many different types of Inverse Eigen value problems and despite of a great deal of research effort being put into this topic many of them are still open and are hard to be solved. In this dissertation ,Yang (2006), proposed optimization algorithms for solving two types of inverse Eigen value problems, namely, the static output feedback problems and the nonnegative inverse Eigen value problems.

In Kressner (2009), a block Newton method has been proposed as a block analog of the nonlinear inverse iteration. Block methods avoid the difficulties with reconvergence by computing all Eigen values in a cluster simultaneously. Unfortunately, the local convergence of these methods seems to be more restricted than that of their single-vector counterparts.

It is known that Eigen value and Eigen vectors play an important role in Graph theory. Regular graphs with small second largest Eigen value have more ’round’ shape, i.e. smaller diameter and higher connectivity. Moreover, not necessarily regular, but sparse graph having strong connectivity properties is known as an expander. Such graphs are relevant to theoretical computer science, the designs of robust computer networks and the theory of error-correcting codes and to complexity theory. Though expanding properties of regular graphs can be measured in several different ways, their common property is a large spectral gap (the difference between the degree and the second largest Eigen value). Koledin and Stani (2013), have obtained a number of regular graphs with small second largest Eigen value (consequently, large spectral gap) and therefore they may be interesting for the application in the above mentioned areas of research.

Eigen value problems also arise in optimization problem. Yuan and Zhang (2013), considered the sparse Eigen value problem, which is to extract dominant (largest) sparse Eigenvectors with at most *k* non-zero components. They proposed a simple yet effective solution called truncated power method that can approximately solve the underlying non-convex optimization problem. A strong sparse recovery result is proved for the truncated power method. The proposed method is tested on applications such as sparse principal component analysis and the densest *k*-sub graph problem.

Random Eigen value problems (REPs) are concerned with determining the probabilistic characteristics of Eigen values and Eigen vectors of random matrices [Forrester (2003)]. First introduced by Wishart in 1928, random matrices are the matrices that are completely defined by statistical distributions. The study of random matrices gained prominence in the 1950, spurred by Wigner's pioneering (1950) work in nuclear physics. The mathematical foundation of the random matrix theory was later established in a series of landmark papers by Wigner (1957), Mehta (1960) and Dyson (1962). A comprehensive account of random matrices can be found in Mehta's seminal work (2004). Bai and Silverstein (2006) established a universality property for the local Eigen value statistics for random matrices. Their main theorem is stated in a much more general setting, and can be applied to various other models of random matrices (such as random real symmetric matrices, for example).

Florian, Schmitt and Rothlauf (2001) have performed a more detailed analysis of the convergence rate using Markov chains. But they considered Genetic Algorithm (GA) in the Markov chain model.

It is known that the Eigen value or spectral gap of a Markov chain is the difference between the two largest Eigen values of the transition matrix of its underlying (state space) graph. McNew (2011) explored the intimate relationship between the spectral gap of a Markov chain and its mixing time, as well as another closely related structural property of a Markov chain known as conductance. The relationships among these properties can be used to put bounds on a chain's mixing time, and can be used to prove both rapid and slow mixing. Vaibhav

(2013) developed new computational methods for solving a general random Eigen value problem (REP) commonly encountered in modeling and simulation of high-dimensional, complex dynamic systems.

Gade (2009) developed a method that uses only time correlation from samples produced from the Markov chain itself. Gade (2009), in his thesis, proposed a novel Krylov subspace type method to estimate the second largest Eigen value from the simulation data of the Markov chain using test functions which are known to have good overlap with the slowest mode. This method starts with the naive Rayleigh quotient estimate of the test function and refines it to obtain an improved estimate of the second largest Eigen value. He applied the method to a few model problems and the estimate compares very favorably with the known answer. He also applied the estimator to some Markov chains occurring in practice, most notably in the study of glasses. He showed experimentally that his estimator is more accurate and stable for these problems compared to the existing methods.

CHAPTER III

Review of Eigen value and Eigenvector and Its Properties

3.1 Introduction

Before discussion of our proposed method and lemma, it is relevant to introduce some definitions, properties and theorems of Eigen value problems. In this aspect we will review the preliminaries of Eigen values and Eigen vectors.

3.2 Some important definitions and associate properties of Eigen value

As in Chapter I we have already discussed about the definition of Eigen value and Eigen vector along with the characteristics polynomial. For the completeness of the discussion we again recall them here. Moreover here we also introduce some important related definitions as well as the properties of Eigen values and Eigen vectors.

Let **A** be a linear transformation matrix of order $n \times n$ and if there is a vector $\mathbf{x} \in \boldsymbol{R^n}$, $\mathbf{x} \neq 0$ and some scalar λ such that

$$\mathbf{Ax} = \lambda\mathbf{x} \tag{3.1}$$

$$\text{or } (\mathbf{A} - \lambda\mathbf{I})\mathbf{x} = \mathbf{0} \tag{3.2}$$

The equation (3.2) has nontrivial solution i.e. $\mathbf{x} \neq 0$ iff

$$P_A(\lambda) = |\mathbf{A} - \lambda\mathbf{I}| = \mathbf{0} \tag{3.3}$$

Here **I** is the identity matrix and **0** is null matrix. Then

(i) **Eigen value problem**: The equation (3.1) or (3.2) forms an algebraic Eigen value problem.

(ii) **Eigen value:** λ is called the Eigen value of **A**.

(iii) **Eigen vector** : Vector **x** is called Eigen vector of **A**. The terms characteristic value and characteristic vector (or proper value and proper vector) are sometimes used instead of Eigen value and Eigen vector.

(iv) **Characteristics equation***:* The equation (3.3) i.e. $|\mathbf{A} - \lambda\mathbf{I}| = 0$ forms a polynomial of degree n (i.e $P(\lambda)$) which is called characteristics equation. The determinant criterion (3.3) is very important because it tells us where to look for the characteristic values of **A**. Note that (a) If $\deg(P_A) \geq 5$ no explicit formula $\longrightarrow$ numerical algorithm required. Also note that in this case find λ by using $P_A(\lambda)$ numerical unstable.

(v) **Eigen system:** The set of all Eigen vectors of a matrix (or linear operator), each paired with its corresponding Eigen value, is called the Eigen system of that matrix. Note that any multiple of an Eigen vector is also an Eigen vector, with the same Eigen value. An Eigen space or characteristic space of a matrix **A** is the set of all Eigen vectors with the same Eigen value, together with the zero vectors. An Eigen basis for **A** is any basis for the set of all vectors that consists of linearly independent Eigen vectors of **A**. Not every matrix has an Eigen basis, but every symmetric matrix does

(vi) **Spectrum**: The set of Eigen values of A is called the spectrum of **A**.

(vii) **Spectral gap**: The spectral gap is the difference between the modulo of the two largest Eigen values of a matrix **A** (of order n). i.e if $\lambda_i: i = 1,2,\cdots,n$ are Eigen values of **A** and $|\lambda_1| \geq |\lambda_2| \cdots \geq |\lambda_n|$ then the spectral gap of **A** is $|\lambda_1 - \lambda_2|$.

(viii) **Eigen gap:** The Eigen gap of a linear operator **A** is the difference between two successive Eigen values, where Eigen values are sorted in ascending order.

(ix) **Spectral radius or Dominant Eigen value:** The largest (in magnitude) Eigen value of equation (3.1) is called dominant Eigen value of **A** or spectral radius. i.e if $|\lambda_i| \geq |\lambda_j| \ \forall\, j = 1,2,\cdots,n$ then λ_i be the dominant Eigen value. Eigen vector $\mathbf{v_1}$ corresponding to λ_1 is called a dominant Eigen vector.

(x) **Singular values of a matrix**: The singular values are simply the absolute values of the Eigen values of a matrix.

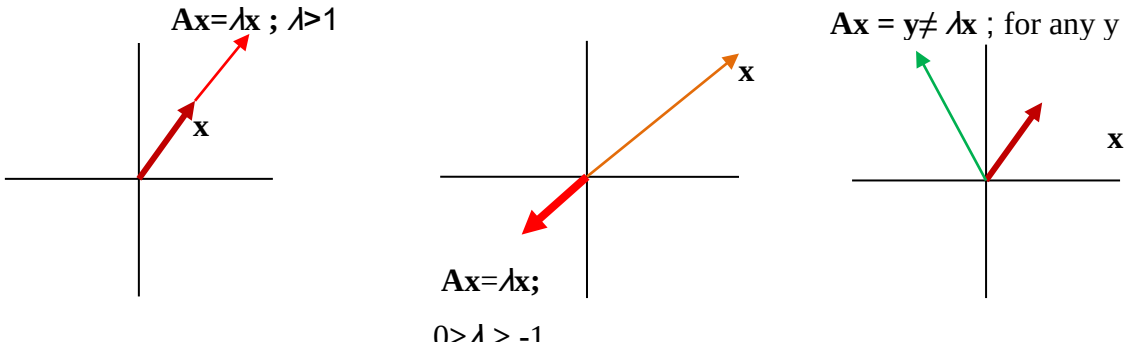
(xi) **Eigen space***:* Let **A** be the matrix representation of a linear operator **T** on a vector space **V**(k). Let λ be an Eigen value of **T** so that $\exists\ \mathbf{v} \neq \mathbf{0}\ \boldsymbol{\epsilon}\ \mathbf{V}$ s.t. **T**(**v**)=**Av** = λ**v**. Then the Eigen space of λ denoted by $\mathbf{V}_\lambda$ is defined as $\mathbf{V}_\lambda = \{\ \mathbf{v}\ \boldsymbol{\epsilon}\ \mathbf{V}\mathbf{:}\ \mathbf{T}(\mathbf{v}) = \lambda\mathbf{v}.\}$. Note that $\mathbf{V}_\lambda$ is a subspace of **V**(k). i.e. the set of all such Eigen vectors $\mathbf{V}_\lambda$ corresponding to each λ is a subspace of **V** called the Eigen space (characteristic space) associated with λ. Note

that if dimension of $\mathbf{V}_\lambda = 1$, then $\mathbf{V}_\lambda$ is called an Eigen line and λ is called a scaling factor).

(xii) **Basis of the Eigen space**: The independent Eigen vectors of $\mathbf{V}_\lambda$ associated with λ form the basis of the Eigen space $\mathbf{V}_\lambda$.

(xiii) **Nullity of the Eigen space**: The number of independent Eigen vectors of $\mathbf{V}_\lambda$ associated with λ form the Nullity of the Eigen space $\mathbf{V}_\lambda$.

(xiv) **Geometry of Eigen value and Eigen vector:** The Eigen value $|\lambda| > 1$ or $|\lambda| = 1$ or $|\lambda| < 1$ or $\lambda = 0$. Geometrically **(a)** $|\lambda| > 1$ implies operator **A** stretch the vector **x** i.e. after operation **A** on **x,** the resulting solution be same in direction (if $\lambda > 1$) (see figure 3.1 (a)) or opposite direction (flipped) (if $\lambda < -1$) but magnitude is increased with $|\lambda|$. **(b)** $|\lambda| < 1$ implies operator **A** shrink the vector **x** i.e. after operation **A** on **x,** the resulting solution be same in direction (if $0 < \lambda < 1$) or opposite direction (flipped) (if $-1 < \lambda < 0$) (see figure 3.1 (b)) but magnitude is decreased with $|\lambda|$. **(c)** $|\lambda| = 1$ implies operator **A** does not stretch or shrink the vector **x** i.e. after operation **A** on **x,** the resulting solution be same in magnitude as well as same in direction (if $\lambda = 1$) or opposite direction (flipped) (if $\lambda = -1$). **(d)** On the other hand if λ = 0, then after operation **A** on **x,** the resulting solution be null vector i.e. **0.**



(a) Stretch positively (b) Shrink but flipped (c) not Eigen vector

Figure 3.1 Geometry of linear transformation and Eigen vector **x**

(xv) **Real matrix:** If $a_{ij}: i = 1,2,\cdots,n$ and $j = 1,2,\cdots,n$ are components of **A** and all components a_{ij} are real the **A** is called real matrix**.**

(xvi) **Trace of the matrix:** If $a_{ij}: i = 1,2,\cdots,n$ and $j = 1,2,\cdots,n$ are components of **A** then $\sum_{i=1}^{n} a_{ii}$ is called trace of the matrix **A**.

(xvii) **Smallest Eigen value:** The Eigen value which is smallest in magnitude is called smallest Eigen value.

(xviii) **Orthogonal matrix:** If $\mathbf{A}^{/} = \mathbf{A}^{-1}$ provided inverse exist, the **A** is called orthogonal matrix**.** i.e. if $\mathbf{A}\mathbf{A}^{/} = \mathbf{A}^{/}\mathbf{A} = \mathbf{I}$

(xix) **Orthogonal vectors:** Two vector **u** and **v** are orthogonal iff $\mathbf{u} \cdot \mathbf{v}^{/} = \mathbf{0}$**.**

(xx) **Normalized vector:** An Eigenvector **x** is said to be normalized if the coordinate of largest magnitude is equal to unity.

(xxi) **Stochastic matrix***:* If $a_{ij}: i = 1,2,\cdots,n$ and $j = 1,2,\cdots,n$ are components of **A** such that $0 \leq a_{ij} \leq 1$ and represents probability (i.e. real value) then matrix **A** is called stochastic matrix or probability matrix or Markov matrix or transition matrix or substitution matrix. Now if each row sum is equal to 1. i.e. $\sum_{j=1}^{n} a_{ij} = 1 \; \forall \; i$ then **A** is called left stochastic matrix. If each column sum is equal to 1. i.e. $\sum_{i=1}^{n} a_{ij} = 1 \; \forall \; j$ then **A** is called right stochastic matrix. Again if each row sum is equal to 1. i.e. $\sum_{j=1}^{n} a_{ij} = 1 \; \forall \; i$ as well as each row sum is equal to 1. i.e. $\sum_{j=1}^{n} a_{ij} = 1 \; \forall \; i$ then **A** is called double stochastic matrix.

(a) Note that the magnitude of each Eigen value of stochastic matrix is less or equal to one.

(b) There exists at least one Eigen value equal to 1 which implies the state of Makov chain is equilibrium.

(xxii) **Quadratic form: A** vector $\mathbf{X} \neq \mathbf{0} \in \Re^{n\times 1}$ and a matrix $\mathbf{A} \in \Re^{n\times n}$, the scalar function defined by $f(x) = \mathbf{x}^{/}\mathbf{A}\mathbf{x} = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}\, x_i x_j$ is called a quadratic form. A quadratic form is said to be positive definite whenever **A** is a positive definite form if and only if f(x) > 0 for all $\mathbf{0} \neq \mathbf{X} \in \Re^{n\times 1}$. **F**or all $\mathbf{x} \neq \mathbf{0}$, $\mathbf{x}^{/}\mathbf{A}\mathbf{x}$ is called quadratic form of **A.** Note that $\mathbf{x}^{/}\mathbf{A}\mathbf{x}$ be positive non negative or negative.

(xxiii) **Symmetric matrix:** If $\mathbf{A}^{/} = \mathbf{A}$ then matrix **A** is called symmetric matrix; thus $a_{kj} = a_{jk}$. Note that all Eigen values of a symmetric matrix are real. Hilbert and Pascal matrices are symmetric

(xxiv) **Skew symmetric**: If $\mathbf{A}^{/} = -\mathbf{A}$ then matrix **A** is called skew symmetric; thus $a_{ij} = -a_{ji}$ and $a_{ii} = 0$.

(xxv) **Positive definite***:* **If** all $\mathbf{x} \neq \mathbf{0}$, $\mathbf{x}^{/}\mathbf{A}\mathbf{x} > 0$ then **A** is called positive definite. Then all Eigen values of **A** are positive. If **A** is symmetric positive definite, then its Eigen values are positive numbers.

(xxvi) **Semi-positive definite**: **If** all $\mathbf{x} \neq \mathbf{0}$, $\mathbf{x}^{/}\mathbf{A}\mathbf{x} \geq 0$ then **A** is called semi positive definite. Then all Eigen values of **A** are non negative.

(xxvii) **Negative definite and semi definite:** Negative definite and Semi negative definite forms are defined by interchanging the words "negative" and "positive" in the above definitions. i.e if all $\mathbf{x} \neq \mathbf{0}$, $\mathbf{x}^{/}\mathbf{A}\mathbf{x}$ is positive definite (semi definite), then $\mathbf{x}^{/}(-\mathbf{A})\mathbf{x}$ is negative definite (negative semi definite). Then for negative definite (semi definite) all Eigen values of **(-A)** are negative (non positive).

(xxviii) **Indefinite forms**: A quadratic form $\mathbf{x}^{/}\mathbf{A}\mathbf{x}$ is said to be indefinite if the form is positive for some points **x** and negative for others.

(xxix) **Algebraic multiplicity** and **geometric multiplicity**: If an Eigen value λ of the matrix **A** is a root of order M_λ of the characteristic polynomial equation (3.3) of A, then M_λ is called algebraic multiplicity of λ. On the other hand if $\mathbf{m}_\lambda$ be the number of linearly independent Eigenvectors corresponding to λ then $\mathbf{m}_\lambda$be the geometric multiplicity of A corresponding λ. That is geometric multiplicity be the dimension of the Eigen space (or null space).

(xxx) **Gerschgorin Circles:** The Eigen values of $\mathbf{A} \in \mathbb{C}^{n\times n}$ are contained the union $\mathcal{G}_r$ of the n Gerschgorin circles defined by $|z - a_{ii}| \leq r_i$ where $r_i = \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}|, i = 1,2,\cdots,n.$

In other words, the Eigen values are trapped in the collection of circles centered at a_{ii} with radii given by the sum of absolute values in $\mathbf{A}_i$ with a_{ii} deleted. (a) Furthermore, if a union μ *of* k Gerschgorin circles does not touch any of the other n$\longrightarrow k$ circles, then there are exactly *k* Eigen values (counting multiplicities) in the circles in μ (b) Since $\sigma(\mathbf{A}^{/}) = \sigma(\mathbf{A})$, the deleted absolute row sums in can be replaced by deleted absolute column sums So the Eigen values of **A** are also contained in the union $\mathcal{G}_c$ of the circles defined by $|z - a_{jj}| \leq c_j$ where $c_j = \sum_{\substack{i=1 \\ i\neq j}}^{n} |a_{ij}|$ for j = 1, 2, … , n. (c) By

By Combining above we can say that the Eigen values of **A** are contained in the intersection $\mathcal{G}_r \cap \mathcal{G}_c$.

(xxxi) **Defective:** An Eigen value whose geometric multiplicity is less than its algebraic multiplicity is defective.

(xxxii) **Complete system:** The matrix $A \in C^{n\times n}$ has a complete system of Eigenvectors if it has n linearly independent Eigenvectors.

(xxxiii) **Vector Norms:** A vector norm on a vector space **X** is a real valued function on **X**, written $\mathbf{x} \rightarrow \|\mathbf{x}\|$, and having following three properties:

(a) $\|\mathbf{x}\| > 0$ for all nonzero vectors **x.**

(b) $\|a\mathbf{x}\| = |a| \ \|\mathbf{x}\|$ for all vectors **x** and all scalar *a*.

(c) $\|\mathbf{x+y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$ for all vectors **x** and **y**.

On $\boldsymbol{R^n}$ the simplest vector norms are:

(xxxiv) **Manhattan/rectangular/ l_1 vector norm :** $\|\mathbf{x}\|_1 = |x_1| + |x_2| + |x_3| + \ldots\ldots\ldots\ldots\ldots\ldots + |x_n|$

(xxxv) **Euclidian / l_2 vector norm:** $\|\mathbf{x}\|_2 = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}$

(xxxvi) **l_∞ vector norm :** $\|\mathbf{x}\|_\infty = \max \{ |x_1|, |x_2|, |x_3| \ldots\ldots\ldots\ldots\ldots\ldots, |x_n| \}$

Here x_i denotes the *i* th component of the vector **x**. For obvious reasons, the l_1 norm is sometimes called the column sum norm and the l_∞ norm is called the dominant element of the vector. Any norm can be thought of as assigning a length to each vector. It is the Euclidean norm that corresponds directly to our usual concept of length, but other norms are sometimes much more convenient for our purposes.

(xxxvii) Let *k* will denote the field of real or complex numbers. Let $k^{m\times n}$ denote the vector space containing all matrices with m rows and n columns with entries in *k*. throughout A^*, denotes the conjugate transpose of matrix **A**. A matrix norm is a vector norm on $k^{m\times n}$. That is, if $\|A\|$ denotes the norm of the matrix **A** then

- $\|\mathbf{A}\| > 0$
- $\|\mathbf{A}\| = 0$ iff $\mathbf{A} = 0$

- $\|\alpha\mathbf{A}\| = |\alpha|\,\|\mathbf{A}\|$ for all α in K and all matrices $\mathbf{A}$ in $k^{m\times n}$
- $\|\mathbf{A}+\mathbf{B}\| \le \|\mathbf{A}\| + \|\mathbf{B}\|$ for all matrices $\mathbf{A}$ and $\mathbf{B}$ in $k^{m\times n}$

Additionally, in the case of square matrices (thus, $m = n$), some (but not all) matrix norms satisfy the following condition, which is related to the fact that matrices are more than just vectors:

- $\|\mathbf{AB}\| \le \|\mathbf{A}\|\,\|\mathbf{B}\|$ for all matrices $\mathbf{A}$ and B in $k^{m\times n}$

A matrix norm that satisfies this additional property is called a sub multiplicative norm.

(xxxviii) **Induced Matrix Norms:** When we solve a system of linear equations **Ax** = **b**, numerically, we shall want to know (among other things) how big the residual vector is? That is conveniently measured by $\|\mathbf{AX-b}\|$, where some norm has been specified. When a vector norm has been specified on R^n ,there is a standard way of introducing a related matrix norm for $n\times n$ matrices; namely,

$$\|\mathbf{A}\| = \sup\{\|\mathbf{Ax}\| : x \in R^n,\ \|\mathbf{x}\| \le 1\}$$

We say that this matrix norm is the subordinate norm to the given vector norm of the norm induced by the given vector norm. The close relationship between the two is useful, because it leads to the following inequality, true for all vectors **x**:

$$\|\mathbf{Ax}\| \le \|\mathbf{A}\|\,\|\mathbf{x}\|$$

The matrix norms subordinate to the vector norms discussed above are respectively;

(xxxix) $\mathbf{l_1}$ **matrix norm**: $\|\mathbf{A}\|_1 = \max\limits_{1\le i\le m} \sum_{i=1}^{m} |a_{ij}|$, which is simply the maximum absolute column sum of the matrix.

(xl) $\mathbf{l_\infty}$ **matrix norm** : $\|\mathbf{A}\|_\infty = \max\limits_{1\le i\le m} \sum_{i=1}^{m} |a_{ij}|$, which is simply the maximum absolute row sum of the matrix.

Here σ_k are the singular values of **A**. Note from above that the matrix norm subordinate to the Euclidean vector norm is not what most students think that is should be; namely:

(xli) **Frobenius norm:** $\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}^2}$.

(xlii) **Spectral norm:** The natural norm induced by the l_2-norm. Let $\mathbf{A}^{\mathrm{H}}$ be the conjugate transpose of the square matrix **A**, so that $(a_{ij})^{\mathrm{H}} = (a_{ji})$, then the spectral norm is defined as the square root of the maximum Eigen value of $\mathbf{A}^{\mathrm{H}}\mathbf{A}$, i.e.,

(xliii) $\|\mathbf{A}\|_2 =$ (maximum Eigenvalue of $\mathbf{A}^{\mathrm{H}}\mathbf{A}$ $)^{1/2}$

$$= \max_{|x|_2 \neq 0} \frac{|\mathbf{Ax}|_2}{|x|_2},$$

This is indeed a matrix norm; however, it is not the one induced by that Euclidean vector norm.

(xliv) Rayleigh Quotient: Let $A \in \mathcal{R}^{n\times n}$ is real and symmetric matrix. The Rayleigh quotient is defined a $r(\mathbf{x}) = \frac{\mathbf{x^T Ax}}{\mathbf{x^T x}}$.

If $(\lambda, \mathbf{v}) \in \mathbb{R}^{n\times n}$ is an Eigen pair, then $r(\mathbf{v}) = \frac{\mathbf{V^T AV}}{\mathbf{V^T V}} = \frac{\mathbf{V^T \lambda V}}{\mathbf{V^T V}} = \boldsymbol{\lambda}\frac{\mathbf{V^T V}}{\mathbf{V^T V}} = \lambda$

3.3 Some important definitions associate with Differential Equations

If $\mathbf{A}_{n\times n}$ is diagonalizable with $\sigma(\mathbf{A}) = \{\lambda_1, \lambda_2, \lambda_3, \dots\dots\dots, \lambda_k\}$, then the unique solution of $\mathbf{u}' = \mathbf{Au}$, $\mathbf{u(0)} = \mathbf{c}$, is given by

$$\mathbf{u} = \mathbf{e^{At}c} = \mathbf{e^{\lambda_1 t}v_1} + \mathbf{e^{\lambda_2 t}v_2} + \cdots + \mathbf{e^{\lambda_k t}v_k}$$

in which $\mathbf{v_i}$ is the Eigen vector $\mathbf{v_i} = \mathbf{G_i c}$, where $\mathbf{G_i}$ is the $\boldsymbol{i^{th}}$ spectral projector (Eigen value). Non homogeneous systems as well as the non diagonalizable case treated are in Example:

Stability

Let $\mathbf{u}' = \mathbf{Au}$, $\mathbf{u(0)} = \mathbf{c}$, where **A** is diagonalizable with Eigen values $\boldsymbol{\lambda_i}$

- If Re ($\boldsymbol{\lambda_i}$) < 0 for each $\boldsymbol{i}$, $\lim_{t\to\infty} e^{At}$, then $= \mathbf{0}$ and $\lim_{t\to\infty} u(t) = \mathbf{0}$, for every initial vector **c.** In this case $\mathbf{u}' = \mathbf{Au}$ is said to be a **stable system,** and **A** is called a **stable** matrix.
- If Re ($\boldsymbol{\lambda_i}$) > 0 for some i , then component of $\mathbf{u}(t)$ can become unbounded as $\mathbf{t} \to \infty$, in which case the system $\mathbf{u}' = \mathbf{Au}$ as well as the underlying matrix **A** are said to be **unstable**.

- If Re $(\lambda_i) \leq 0$ for each i **,** then component of $\mathbf{u}(t)$ remain finite for all t, but some can oscillate indefinitely. This is called a **semi stable** situation.

3.4 Some important theorems and associate properties of Eigen values

In previous section we recalled some important definitions and associate definitions regarding Eigen value problems. In this section we will introduces some important theorems and associate properties regarding Eigen value problems. For proofs of the theorems, reader is requested to see any standard linear algebra book.

Theorem (i): The equation $\mathbf{Au} = \lambda\mathbf{u}$ has a non-trivial solution **u** iff λ is a latent root (Eigen value) of **A**. i.e. λ is a latent root of a matrix **A** iff it is a root of the characteristic equation of **A**. i.e. The Eigen values of a linear transformation **A** are the scalars λ which satisfy the equation $|\mathbf{A}-\lambda\mathbf{I}| = 0$.

Theorem (ii): **Fundamental theorem of Algebra:** Every polynomial over the complex field **C** has a root. Where $\mathbf{M}=\mathbf{A}-\lambda\mathbf{I}$, that is, **M** is obtained by subtracting λ down the diagonal of **A**. Note that some matrices have no Eigen values and hence no Eigenvectors.

Theorem (iii):The scalar λ is a characteristic root of the matrix **A** iff the matrix $(\mathbf{A}-\lambda\mathbf{I})$ is singular.

Theorem (iv) : If $\lambda_1, \lambda_2, \lambda_3, \ldots\ldots\ldots\ldots, \lambda_k$ are distinct Eigen values (roots) of a matrix **A** and if $\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}, \ldots, \mathbf{x_k}$ are non zero Eigen (invariant) vectors associated respectively with these roots, then $\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}, \ldots, \mathbf{x_k}$ are linearly independently.

Theorem (v) : The *k*th derivative of $p(\lambda) = |\mathbf{A} - \lambda\mathbf{I}|$, where A is *n* square, with respect to λ (a) is *k* ! times the sum of the principal minors of order *n-k* of the characteristic matrix when $k<n$, (b) is *n*! when k = n, and (c) is 0 when $k>n$.

Theorem (vi) : If λ_i is an r –fold characteristic roots (Eigen value) of an n-square matrix **A**, the rank of $(\mathbf{A} - \lambda\mathbf{I})$ is not less than n-r and the dimension of the associated invariant vector space is not greater than r.

Theorem (vii) : The characteristic roots of **A** and $\mathbf{A}^{/}$are the same.

Theorem (viii) :Since any principal minor of $\overline{\mathbf{A}}^{/}$ is the conjugate of the corresponding principal minor of **A**, the characteristic roots of $\overline{\mathbf{A}}$ are the conjugates of the characteristic roots of **A**.

Theorem (ix) : If $\lambda_1, \lambda_2, \lambda_3, \ldots\ldots\ldots\ldots\ldots, \lambda_n$ are the characteristic roots of an n-square matrix **A** and if k is a scalar, then $k\lambda_1, k\lambda_2, k\lambda_3, \ldots\ldots\ldots\ldots\ldots, k\lambda_n$ are the characteristic roots of $(k\mathbf{A})$.

Theorem (x): If $(\lambda_i,\mathbf{x})$ be the Eigen pair of an n-square matrix **A** and if k is a scalar, then λ_i-k, **x** be the Eigen pair of $(\mathbf{A}-k\,\mathbf{I})$.

Theorem (xi): If $(\alpha,\mathbf{x})$ be the Eigen pair of a non-singular matrix **A**, then $|A|/\alpha$, **x** be the Eigen pair of adj **A**.

Theorem (xii): $(\alpha,\mathbf{x})$ be the Eigen pair of a non-singular matrix **A**, then $1/\alpha$, **x** be the Eigen pair of $\mathbf{A}^{-1}$.

Theorem (xiii): If **x** be Eigen vector associated with Eigen value λ of a matrix **A**, then **x** be remained Eigen vector of the matrix $(k\mathbf{A})$, $(\mathbf{A}-k\mathbf{I})$, $\mathbf{A}^{-1}$ and $(k\mathbf{A}^{-1})$ associated with Eigen value $k\lambda$, $\lambda-k$, $1/\lambda$ and k/λ respectively.

Theorem (xiv): Assume that the $n\times n$ matrix **A** has distinct Eigen values $\lambda_1, \lambda_2, \lambda_3, \ldots\ldots\ldots\ldots\ldots, \lambda_n$ and consider the Eigen value λ_j . Then a constant α can be chosen so that $\mu_1 = 1/(\lambda_j - \alpha)$ is the dominant Eigen value of $(A-\alpha\mathbf{I})^{-1}$.

Theorem (xv): A symmetric matrix $\mathbf{A} \in \mathcal{R}^{n\times n}$ is positive definite if and only if all of its Eigen values are positive.

Theorem (xvi): Geometrically, **A** does not rotate any vector by more than $\pi/2$.

Theorem (xvii): If **A** be square matrix with Eigen values $\lambda_1, \lambda_2, \lambda_3, \ldots\ldots\ldots\ldots\ldots, \lambda_n$ then

$$tr(\mathbf{A}) = \sum_{i=1}^{n} a_{ii} = \sum_{i=1}^{n} \lambda_i = \lambda_1 + \lambda_2 + \lambda_3 + \cdots + \lambda_n$$

Theorem (xviii): The determinant of **A** is the product of all Eigen values:

$$\det(\mathbf{A}) = \prod_{i=1}^{n} \lambda_i = \lambda_1 \lambda_2 \ldots \lambda_n$$

Theorem (xix): The Eigen values of the *k*th power of **A**, i.e. the Eigen values of $\mathbf{A}^k$, for any positive integer *k*, are $\lambda_1^k, \lambda_2^k, \lambda_3^k, \ldots, \lambda_n^k$

Theorem (xx): The matrix **A** is invertible if and only if all the Eigen values λ_i are nonzero. If **A** is invertible, then the Eigen values of $\mathbf{A}^{-1}$ are $\frac{1}{\lambda_1}, \frac{1}{\lambda_2}, \ldots, \frac{1}{\lambda_n}$, . Clearly, the geometric multiplicities coincide. Moreover, since the characteristic polynomial of the inverse is the reciprocal polynomial for that of the original, hey share the same algebraic multiplicity.

Theorem (xxi): If **A** is equal to its conjugate transpose $\mathbf{A}^*$(in other words, if **A** is Hermitian), then every Eigen value is real. The same is true of any a symmetric real matrix. If **A** is also positive-definite, positive-semi definite, negative-definite, or negative-semi definite every Eigen value is positive, non-negative, negative, or non-positive respectively.

Theorem (xxii): Every Eigen value of a unitary matrix has absolute value $|\lambda| = 1$.

Theorem (xxiii): Let **A** be an arbitrary $n \times n$ matrix of complex numbers with Eigen values $\lambda_1, \lambda_2, \lambda_3, \ldots\ldots\ldots\ldots\ldots, \lambda_n$ (Here it is understood that an Eigen value with algebraic multiplicity μ occurs μ times in this list.) Then $\mathrm{x}^{\mathrm{T}}\mathrm{Ax} > 0$ for every nonzero $\mathbf{x} \in \mathcal{R}^{n \times 1}$ (most commonly used as the definition).

(a) All Eigen values of **A** are positive.

(b) $\mathbf{A} = \mathbf{B}'\mathbf{B}$ for some nonsingular **B.** While **B** is not unique.

(c) But there is one and only one upper-triangular matrix **R** with positive diagonals such that $\mathbf{A} = \mathbf{R}'\mathbf{R}$ This is the Cholesky factorization of **A.**

(d) The leading principal minors of **A** are positive.

(e) All principal minors of **A** are positive.

Theorem (xxiv): If x_0 is chosen appropriately, then the sequences

$\left\{\mathbf{X}_k = \left(\mathbf{x}_1^{(k)}, \mathbf{x}_2^{(k)}, \ldots, \mathbf{x}_n^{(k)}\right)^{\mathrm{T}}\right\}$ and $\{C_k\}$ generated recursively by $\mathbf{Y}_k = (\mathbf{A} - \alpha\mathbf{I})^{-1}\mathbf{x}_k$ and $\mathbf{X}_{k+1} = \frac{1}{c_{k+1}}\mathbf{Y}_k$ where $c_{k+1} = x_j^{(k)}$ and $x_j^{(k)} = \max_{1 \le i \le n}\left\{\left|x_i^{(k)}\right|\right\}$, will converge to the dominant Eigen pair (μ_1, $\mathbf{x}_1$) of the matrix $(\mathbf{A}\text{-}\mathbf{kI})^{-1}$. Finally, the corresponding Eigen value for the matrix **A** is given by the calculation $\lambda_j = \left(\frac{1}{\mu 1}\right) + \alpha$

Theorem (xxv): To compute the dominant Eigen value λ_1 and its associated Eigen vector $\mathbf{x_1}$ for the $n \times n$ matrix **A**. It is assumed that the n Eigen values have the dominance property.

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \ldots\ldots |\lambda_n|.$$

Theorem (xxvi): If **A** is diagonalizable, then **A** has *n* linearly independent Eigen vectors with corresponding Eigen values. Note that since *n* Eigen vectors are linearly independent, they must form a basis of Eigen space.

3.5 Iterative methods

In solving an Eigen value problem by iterative methods, there are a number of properties and conditions that need to be considered. These greatly affect the choice of algorithm. We list below a number of questions that an investigator needs to consider in solving a particular problem.

(i) Is the matrix real or complex?

(ii) What special properties does the matrix have? Is it

(a) symmetric?

(b) Hermitian?

(c) Skew symmetric?

(d) Unitary?

(e) Structure?

(f) Band?

(g) Sparse?

(h) Structured sparseness?

(i) Toeplitz? and so on

(iii) Which Eigen value required?

(a) Largest in magnitude?

(b) Smallest in magnitude?

(c) Real part of Eigen values?

(d) Nature of sign of Eigen values?

(e) Sums of intermediate Eigen values?

(f) Real part of Eigen values?

There are many iterative numerical methods which deal with specific Eigen value and Eigen vectors along with specific form of matrix. Below we have listed few well-known methods with specific purposes

i. **Power method:** Power method for single largest (in magnitude) Eigen value and corresponding Eigen vector.

ii. **Inverse Power method:** Inverse Power method for single smallest (in magnitude) Eigen value closest to zero and corresponding Eigen vector.

iii. **Inverse Power method with shift:** Inverse Power method with shift for desired single Eigen pairs.

iv. **Rayleigh quotient iteration:** Rayleigh quotient iteration is fast to converse for single Eigen pairs.

v. **Simultaneous iteration:** Simultaneous iteration for many Eigen values

vi. **QR algorithm:** QR algorithm (with shift) for all Eigen values (fast) but not find out corresponding Eigen vectors.

3.6 Power method

In mathematics, the power iteration is an Eigen value algorithms: given a matrix **A,** the algorithm will produce a number λ (the Eigen value) and a nonzero vector **v** (the Eigenvector), such that $\mathbf{Av} = \lambda\mathbf{v}$. The Power iteration is a very simple algorithm. It does not

compute matrix decomposition and hence it can be used when **A** is a very large sparse matrix. Like the Jacobi and Gauss-Seidel methods, the power method for approximating Eigen values is iterative. First assume that the matrix **A** has a dominant Eigen value with corresponding dominant Eigen vectors (Under the assumptions: **A** has an Eigen value that is strictly greater in magnitude than its other Eigen values). The starting vector has a nonzero component in the direction of an Eigen vector associated with the dominant Eigen value. Then choose an initial approximation Eigen vector of **A** say $\mathbf{x}_0$, which may be an approximation to the dominant Eigen vector or a random vector. This initial approximation must be a non zero vector. The method is described by the iteration. So, at every iteration, the vector $\mathbf{x}_k$ is multiplied by the matrix **A** and normalized. Then: **A** subsequence of converges to an Eigenvector associated with the dominant Eigen value. Note that the sequence does not necessarily converge. Finally, the sequence is given by

$$\mathbf{x}_1 = \mathbf{A}\mathbf{x}_0$$
$$\mathbf{x}_2 = \mathbf{A}\mathbf{x}_1 = \mathbf{A}(\mathbf{A}\mathbf{x}_0) = \mathbf{A}^2\mathbf{x}_0$$
$$\mathbf{x}_3 = \mathbf{A}\mathbf{x}_2 = \mathbf{A}(\mathbf{A}^2\mathbf{x}_0) = \mathbf{A}^3\mathbf{x}_0$$
$$\vdots$$
$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} = \mathbf{A}(\mathbf{A}^{k-1}\mathbf{x}_0) = \mathbf{A}^k\mathbf{x}_0$$

For large powers of *k*, and by properly scaling this sequence, we will see that we obtain a good approximation of the dominant Eigen vector of **A**.

Algorithm of Power Method

1: Choose a starting vector $x^{(0)} \in F^n$ with $\|x^{(0)}\| = 1$.
2: k=0.
3: repeat
4: k :=k+1;
5: $y^{(k)} := \mathbf{A}\mathbf{x}^{(k-1)}$;
6: $\mu_k := \|y^{(k)}\|$;
7: $x^{(k)} := \mathbf{y}^{(k)}/\mu_k$;
8: until a convergence criterion is satisfied

The vector $x^{(k)}$ generated by algorithm have all norm (length) one. That is $\left\{\mathbf{x}^{(k)}\right\}_{k=0}^{\infty}$ is a sequence on the until sphere in F^n.

Theorem : If **A** is an $n \times n$ diagonalizable matrix with a dominant Eigen value, then there exist nonzero $\mathbf{x_0}$ such that the sequence of vectors given by

$$\mathbf{Ax_0},\ \mathbf{A^2x_0},\ \mathbf{A^3x_0},\ \mathbf{A^4x_0},\ \cdots \mathbf{A}^{\mathrm{k}}\mathbf{x_0}, \cdots$$

approaches a multiple of the dominant Eigen vectors of **A.**

Proof: Because **A** is diagonalizable, then **A** has n linearly independent Eigen vectors with corresponding Eigen values. Assume that these Eigen values are ordered so that λ_1 is the dominant Eigen value (with a corresponding Eigen vectors of $\mathbf{x}_1$). Let **A** be an n × n matrix with Eigen values $\lambda_1,\ \lambda_2,\ \lambda_3,\ \ldots\ , \lambda_n$, not necessarily distinct, that satisfy the relations $|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_n|$. The Eigen value$\lambda_1$, which is largest in magnitude, is known as the dominant Eigen value of the matrix **A**. Furthermore, assume that the associated Eigen vectors $\boldsymbol{\nu_1, \nu_2, \nu_3, \ldots\ , \nu_n}$ are linearly independently, and therefore form a basis for $\mathbf{R}^n$. It should be noted at this point that not all matrices have Eigen values and Eigen vectors which satisfy the conditions we have assumed here.

Let $\mathbf{x}^{(0)}$ be a nonzero element of $\mathbf{R}^n$. Since the Eigenvectors of **A** form a basis for $\mathbf{R}^n$, it follows that $\mathbf{x}^{(0)}$ can be written as a linear combination of $\boldsymbol{\nu_1, \nu_2, \nu_3, \ldots\ , \nu_n}$; that is, there exist contains $\boldsymbol{\sigma_1, \sigma, \sigma_3, \ldots\ , \sigma_n}$ such that

$$\mathbf{x}^{(0)} = \alpha_1\boldsymbol{\nu_1} + \boldsymbol{\alpha_2\nu_2} + \boldsymbol{\alpha_3\nu_3} + \ \ldots + \alpha_{\mathbf{n}}\boldsymbol{\nu_n}\ .$$

Next, construct the sequence of vectors $\left\{\mathbf{x}^{(m)}\right\}$ according to the rule $\mathbf{x}^{(m)} = \mathbf{Ax}^{(m-1)}$ for m ≥ 1. By direct calculation we find

$$\begin{aligned}
\mathbf{x}^{(1)} &= A\mathbf{x}^{(0)} = \alpha_1(A\nu_1) + \alpha_2(A\nu_2) + \alpha_3(A\nu_3) + \ \ldots + \alpha_n(A\nu_n) \\
&= \alpha_1(\lambda_1\boldsymbol{\nu}_1) + \alpha_2(\lambda_2\boldsymbol{\nu}_2) + \alpha_3(\lambda_3\boldsymbol{\nu}_3) + \ \ldots + \alpha_n(\lambda_n\boldsymbol{\nu}_n) \\
\mathbf{x}^{(2)} &= A\mathbf{x}^{(1)} = A^{(2)}\mathbf{x}^{(0)} \\
&= \alpha_1(A^2\boldsymbol{\nu}_1) + \alpha_2(A^2\boldsymbol{\nu}_2) + \alpha_3(A^2\boldsymbol{\nu}_3) + \ \ldots + \alpha_n(A^2\boldsymbol{\nu}_n) \\
&= \alpha_1\left(\lambda_1{}^2\boldsymbol{\nu}_1\right) + \alpha_2\left(\lambda_2{}^2\boldsymbol{\nu}_2\right) + \alpha_3\left(\lambda_3{}^2\boldsymbol{\nu}_3\right) + \ \ldots + \alpha_n(\lambda_n{}^2\boldsymbol{\nu}_n)
\end{aligned}$$

and, in general,

$$\mathbf{x}^{(m)} = A\mathbf{x}^{(m-1)} = \cdots \boldsymbol{A}^{(m)}\,\mathbf{x}^{(0)}$$
$$= \alpha_1(A^m\boldsymbol{\nu}_1) + \alpha_2(A^m\boldsymbol{\nu}_2) + \alpha_3(A^m\boldsymbol{\nu}_3) + \ \ldots + \alpha_n(A^m\boldsymbol{\nu}_n)$$
$$= \alpha_1({\lambda_1}^m\boldsymbol{\nu}_1) + \alpha_2({\lambda_2}^m\boldsymbol{\nu}_2) + \alpha_3({\lambda_3}^m\boldsymbol{\nu}_3) + \ \ldots + \alpha_n({\lambda_n}^m\boldsymbol{\nu}_n)$$

In deriving these expressions we have made repeated use of the relation $A\boldsymbol{v}_j = \boldsymbol{\lambda}_j\boldsymbol{v}_j$, which follows from the fact that $\boldsymbol{v}_j$ is an Eigenvector associated with the Eigen value $\boldsymbol{\lambda}_j$.

Factoring ${\lambda^m}_1$ from the right –hand side of the equation for $\boldsymbol{x}^{(m)}$ gives

$$\mathbf{x}^{(m)} = {\lambda_1}^m\left[\alpha_1\boldsymbol{\nu}_1 + \alpha_2\left(\frac{\lambda_2}{\lambda_1}\right)^m\boldsymbol{\nu}_2 + \alpha_3\left(\frac{\lambda_3}{\lambda_1}\right)^m\boldsymbol{\nu}_3 + \ \ldots\ \alpha_n\left(\frac{\lambda_n}{\lambda_1}\right)^m\boldsymbol{\nu}_n\right].$$

By assumption, $\left|\frac{\lambda_j}{\lambda_1}\right| < 1$ for each *j*, So $\left|\frac{\lambda_j}{\lambda_1}\right|^m \to 0\ as\ m \to \infty$. So

$$\lim_{m\to\infty}\frac{\mathbf{x}^{(m)}}{{\lambda_1}^m} = \alpha_1\boldsymbol{\nu_1}.$$

Since any nonzero constant times an Eigenvector is still an Eigen vector associated with the same Eigen value, We see that the scaled sequence $\{\mathbf{x}^{(m)}/{\lambda_1}^m\}$ converges to an Eigen vector associated with the dominant Eigen value provided$\alpha_1 \neq 0$. Furthermore, convergence toward the Eigen vector is linear with asymptotic error constant $\left|\frac{\lambda_2}{\lambda_1}\right|$.

To simplify the notation, let's introduce the vector $\boldsymbol{y}^{(m)}$ to denote the result of multiplying by the matrix **A***;* that is $\boldsymbol{y}^{(m)} = A\boldsymbol{x}^{(m-1)}$, $\boldsymbol{x}^{(m)}$ is then calculated by the formula

$$\boldsymbol{x}^{(m)} = \frac{y^{(m)}}{{y_{p_m}}^{(m)}},$$

where p_m is an integer chosen so that $\left|{y_{p_m}}^{(m)}\right| = ||y^{(m)}||_\infty$. Note that p_m is an index into the vector $\boldsymbol{y}^{(m)}$.

To avoid overflow and underflow problems when calculating the sequence $\{\mathbf{x}^{(m)}\}$ (note that $\lim_{m\to\infty}{\lambda_1}^m \to \pm\infty$ when $|\lambda_1| > 1$. where as $\lim_{m\to\infty}{\lambda_1}^m \to 0$ when $|\lambda_1| < 1$, It is common practice to scale the vectors $\boldsymbol{x}^{(m)}$ so that they are all of unit length. Here, we will use the $l_\infty - norm$ to measure vector length. Thus in a practical implementation of the Power method, the vector $\boldsymbol{x}^{(m)}$ would be computed on two steps: First multiply the previous vector by the matrix **A** and then scale the resulting vector to unit length.

The proof of the Theorem provides some insight into the rate of convergence of the Power method. That is, if the Eigen values of **A** are ordered so that

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_n|.$$

then the power method will converge quickly if $\left|\frac{\lambda_2}{\lambda_1}\right|$ is small, and slowly if is close to 1. The rate of convergence is displayed in the figure 3.2 (for an example).

Then the power method will converge quickly if $|\lambda_2|/|\lambda_1|$ is small, and slowly if $|\lambda_2|/|\lambda_1|$ is close to 1. The convergence is illustrated in figure 3.2.

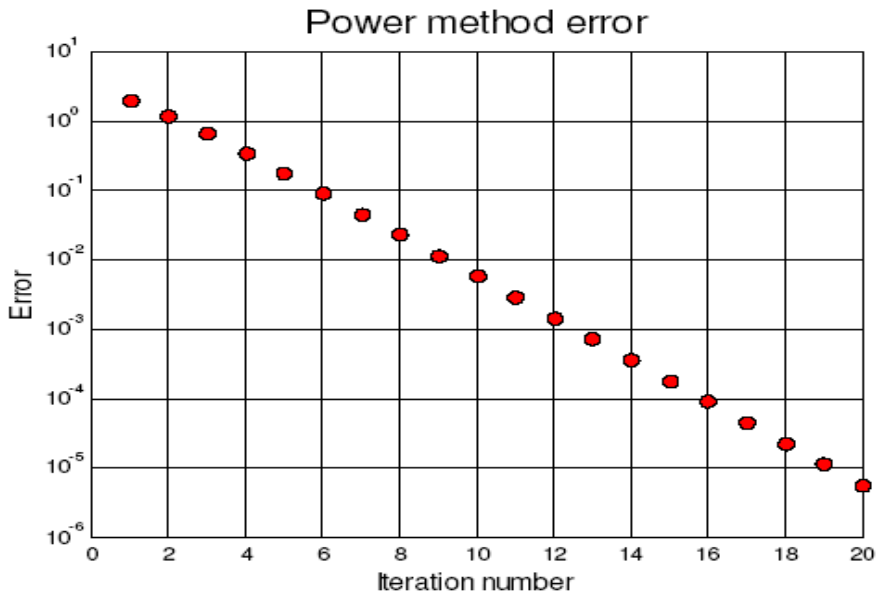


Figure 3.2: Iteration number versus errors in power method

Summary of the Power method :

1. The Power method can be used to find the dominant Eigen value of a symmetric matrix.
2. The method has linear convergence.
3. The method requires an initial guess and it is not obvious how this can be chosen in practice.
4. The method does work if the dominant Eigen value has multiplicity *r*. The estimated Eigen vector will then be a linear combination of the *r* Eigenvectors.

5. The speed of convergence of the Power iteration depends on the ratio of the second largest Eigen value (in absolute value) to the largest Eigen value (in absolute value). The convergence rate of the power method depends on $I\lambda_2/\lambda_1 I$, where λ_1 is the largest Eigen value and λ_2 is the second largest Eigen value of **A** in magnitude. This ratio is generally smaller than1, allowing adequate convergence. But there are cases where this ratio can be very close to 1, causing very slow convergence. i.e. if the ratio of the largest Eigen value and second Eigen value is near to one i.e. if $I\lambda_2/\lambda_1 I \rightarrow 1$ then the speed of convergence be extremely slow.
6. If **z** be the initial guess of Power method and Let $\mathbf{x_1}$ be the Eigen vector corresponding to dominant Eigen value λ_1 i.e $\lambda_1 = \lambda_{max}(\mathbf{A})$. The angle $\measuredangle(\mathbf{z},\mathbf{x}_1)$ between $\mathbf{x_1}$ and **z** is defined by the relation

$$\cos < (z, x_1) = \frac{z^* x_1}{||z||_2 \, ||x_1||_2}.$$

(a) If the starting vector **z** and the Eigen vector $\mathbf{x_1}$ are perpendicular to each other, then$\cos < (z, x_1) = 0$. In this case the Power method does not converge in exact arithmetic.

(b) On the other hand, if , $\cos < (z, x_1) \neq 0$ the Power method generates a sequence of vectors that become increasingly parallel to $\mathbf{x_1}$. This condition on the angle $< (z, x_1)$ is true with very high probability if **z** is chosen at random.

7. The Power method can be used to find dominant Eigen value of a symmetric matrix.
8. The Power method requires an initial guess and it is not obvious how this can be chosen in practice. But random choose has very high probability for convergence of the algorithm.
9. The method does work if the dominant Eigen value has multiplicity *r.* Then the estimated Eigen vector will be a linear combination of the r Eigen vectors.

Although the Power method has worked well in these examples, we must say something about cases in which the Power method may fail. There are basically three such cases:

1. Using the Power method when **A** is not diagonalizable. Recall that **A** has *n* linearly independent Eigen vectors if and only if **A** is diagonalizable. Of course, it is not easy to tell by just looking at **A** whether it is diagonalizable.
2. Using the Power method when **A** does not have a dominant Eigen value, or when the dominant Eigen value is such that

   $$|\lambda_1| > |\lambda_2| \text{ but } |\lambda_1| = |\lambda_2|$$

   Then $|\lambda_1/\lambda_2|$ is barely less than 1, and high powers of $|\lambda_1/\lambda_2|$ do not tend to zero quickly. Again, it is not easy to determine whether **A** has this defect by just looking at **A**.
3. If the entries of **A** contain significant error. Powers A^m of **A** will have significant round off error in their entries.

The Power iteration is a very simple algorithm and elegant. It does not compute a matrix decomposition, and hence it can be used when **A** is a very large sparse matrix. But Power method suffers some major drawbacks. The method only returns a single Eigen vector estimate, and it is always the one corresponding to the Eigen value of largest magnitude. In addition, convergence is only guaranteed if the Eigen values are distinct in particular, the two Eigen values of largest absolute value must have distinct magnitudes. The rate of convergence primarily depends upon the ratio of these magnitudes, so if the two largest Eigen values have similar sizes, then the convergence will be slow. In spite of its drawbacks, the power method is still used in some applications, since it works well on large, sparse matrices when only a single Eigen vector is needed.

3.7 Inverse Power method

The Power method is able to find out only largest Eigen pair. The method could not find other Eigen pairs like smallest Eigen pair. On the other hand Inverse Power method is able to find out smallest Eigen value or desire Eigen value by using shifting Eigen property. Observe that subtracting a constant from the diagonal elements of **A** gives a system whose Eigen values are those of **A** with the same constant subtracted:

Given $\mathbf{Ax} = \lambda\mathbf{x}$.

Subtract $s\mathbf{Ix} = s\mathbf{x}$ from both sides:

$$\mathbf{Ax}-s\mathbf{Ix} = \lambda\mathbf{x} - s\mathbf{x},$$

$$(\mathbf{A}-s\mathbf{I})\mathbf{x} = (\lambda - s)\mathbf{x},$$

This relationship can be applied in two ways. Suppose we wish to determine the value of an Eigen value near to some numbers. We shift the Eigen values by subtracting s from the diagonal elements: there is then an Eigen value very near to zero in the shifted matrix. We use the power method on the inverse of the shifted matrix. This is often rapidly convergent because the reciprocal of the very small value is very large and is usually much larger then the next largest one (for the shifted inverse system). After we obtain it, we reverse the transformations to obtain the desired value for the original matrix. This process is called the Inverse Power Method.

The Eigen value problem is stated as $\mathbf{Ax} = \lambda\mathbf{x}$. If $|\mathbf{A}| \neq 0$, then this equation is reformulated as

$$\mathbf{A}^{-1}x = \frac{1}{\lambda}x,$$

We see that the Eigen value of the matrix $\mathbf{A}^{-1}$ obtained the Power method will be the largest $(1/\lambda)$ or the smallest λ. Hence the Power method applied to the matrix $\mathbf{A}^{-1}$ is called the Inverse Power method and yields the lowest Eigen value by writing.

$$(\mathbf{A}- q\mathbf{I})\,\mathbf{x} = (\lambda - q)\,\mathbf{x}$$

which can be put in the form

$$(\mathbf{A}- q\mathbf{I})^{-1}\mathbf{x} = = \frac{1}{\lambda - q}\,\mathbf{x},$$

By choosing q very close to an Eigen value and applying the Power method on the matrix $(A- qI)^{-1}$ we will converge on the largest value of $\frac{1}{\lambda - q}$ which means that we have converged on the Eigen value closest to q. The iterative procedure can be described in the following way:

$$(\mathbf{A}-q\mathbf{I})^{-1}\,\mathbf{x}^{(i-1)} = \mathbf{y}^{(i)}$$

or $(\mathbf{A}-q\mathbf{I})\,\mathbf{y}^{(i)} = \mathbf{x}^{(i-1)}$

where $x^{(i-1)}$ is an initial guess for the Eigen vector of the Eigen value in the vicinity of the value q; and $\mathbf{y}^{(i)}$ is the improved approximation to the Eigen vector. $\mathbf{y}^{(i)}$ is evaluated by performing Gaussian elimination on the square matrix $(\mathbf{A}-q\mathbf{I})$ using the initial guess vector $\mathbf{x}^{(i-1)}$ as the right hand side of the system. Once $\mathbf{y}^{(i)}$ is evaluated, it is normalized by all its

elements by the element having the largest magnitude. This normalized vector is now used as the new $\mathbf{x}^{(i-1)}$ and the procedure is repeated to find a better approximation $\mathbf{y}^{(i)}$. The element having the largest magnitude in $\mathbf{y}^{(i)}$ which is used to normalize it, is also an approximation to the Eigen value of $(\mathbf{A}\text{-q}\mathbf{l})^{-1.}$ This is equated to $\frac{1}{\lambda i-q}$ to evaluate the Eigen value. This procedure is continued for consecutive Eigen value approximations, until the relative error between the consecutive values satisfies a relative error criterion

The Inverse Power method is more powerful than the Power method, because it allows all the Eigen values to the found by appropriately choosing q. However the method is extremely tedious and computationally expensive since Gaussian elimination must be performed for each iteration.

3.8 Algorithm of Inverse Power method (Inverse vector iteration)

Algorithm of Inverse vector iteration

1: Choose a starting vector $\mathbf{x}_0 \in F^n$ and a shift σ.

2: Compute the LU factorization of $\mathbf{A} - \sigma\mathbf{I}$: **LU=P** $(\mathbf{A} - \boldsymbol{\sigma}\mathbf{I})$

3: $\mathbf{y}^{(0)} := \mathbf{U}^{-1}\mathbf{L}^{-1}\mathbf{P}\mathbf{x}^{(0)}$. $\mu^{(0)} = \boldsymbol{y}^{(0)*}\mathrm{x}^{(0)}$, $\lambda^{(0)} := \sigma + \frac{1}{\mu^{(0)}}$. $k := 0$.

4: while $||\mathbf{x}^{(k)} - \frac{\mathbf{y}^{(k)}}{\mu^{(k)}}|| > \text{tol}\,||\mathbf{y}^{(k)}||$ do

5: $k := k+1$.

6: $\mathbf{x}^{(k)} := \mathbf{y}_{k-1}/||\boldsymbol{y}_{k-1}||$.

7: $\mathbf{y}^{(k)} := \mathbf{U}^{-1}L^{-1}\mathbf{P}\mathbf{x}^{(k)}$.

8: $\mu^{(k)} := \mathbf{y}^{(\mathbf{k})*}\mathbf{x}^{(k)}, \lambda^{(k)} := \sigma + 1/\mu^{(k)}$.

9: end while

where we have used

$$\mathbf{A}\boldsymbol{y}^{(\boldsymbol{k})} - \lambda^{(\boldsymbol{k})}\mathbf{y}^{(\boldsymbol{k})} = \mathbf{A}\mathbf{y}^{(\boldsymbol{k})} - \left(\sigma - \frac{\mathbf{1}}{\boldsymbol{\mu}^{(\boldsymbol{k})}}\right)\boldsymbol{y}^{(\boldsymbol{k})} = \mathbf{x}^{(\boldsymbol{k})} - \boldsymbol{y}^{(\boldsymbol{k})}/\mu^{(\boldsymbol{k})}$$

The convergence of the Theorem can easily be adapted to the new situation if it is taken into account that $\mathbf{A} - \sigma\mathbf{I}$ has Eigen pairs $(\mu_i, \mathbf{u}_i)$ with $\mu_i = \frac{1}{(\sigma-\lambda_i)}$.

3.9 Summary

The Power method and Inverse Power method are two methods for numerical calculation of Eigen values as well as Eigen vectors of real matrices. The Stability of a numerical Eigen value problem depends on the matrix under consideration. If the matrix is symmetric with symmetrically distributed error, then the calculated Eigen values will approximate the actual Eigen values, provided the Eigen values are all simple. Otherwise, the numerical methods may fail to find all Eigen values.

CHAPTER IV

Proposed Modified Hybrid Iterative Algorithm

4.1 Introduction

Many physical situations, related to Eigen value problem, occur in real live in which largest and or smallest Eigen value is required and in some cases nature of the Eigen values may required. Therefore proposed algorithm will be developed to purpose these issues. Actually, the proposed algorithm will be a hybridization of existing Power method and Inverse Power method along with some modifications so that the proposed algorithm is able to find out not only largest and smallest Eigen pairs along with the nature of Eigen value sequences. Therefore, before proposed the modified algorithm, at first, we would like to present existing algorithm of Power method and Inverse Power method in brief.

4.2 Existing Algorithm of Power Method and Inverse Power Method

At first consider an algebraic Eigen value problem

$$\mathbf{Ax} = \lambda \mathbf{x} \tag{4.1}$$

Here **A** is $n \times n$ coefficient matrix of a system of linear equations and $(\lambda_i, \mathbf{x}_i) : i=1,2, \ldots, n$ are Eigen pairs of the system. It is also assumed that **A** is positively defined matrix or at least largest and smallest Eigen values are real and unique.

As mentioned earlier that the proposed algorithm will be developed by hybridization of Power method and Inverse Power method with some modification, so we recalled the pseudo-code of Power method as well as Inverse power method here. In the following pseudo-codes, the Power method is defined as a function and denoted as **Power Method ( )** and **Inverse Power Method** is also defined as a function and denoted as **Inverse Power Method ( )**. The pseudo-codes of both functions are presented below respectively.

Power Method ( ):

```
{
  Step (1): read A                          //read the elements of matrix A
            set  y = x0                     //∋ ||x0|| = 1 the initial guess
            set  ξ= ξ0                      // the relative tolerant error
            set  Imax                       //maximum  number of iteration
            for k = 1,2, ……, Imax do        //Starting loop
              {
  Step (2):      v = y / ||y||2             //Normalization of guess solution
  Step (3):      y = Av                     //Matrix multiplication
  Step (4):      θ = v * y                  //Guess Eigen value
  Step (5):      if ||y – θ v||2 ≤ ξ |θ|,    // Stopping criterion
                 set (λ , x )= (θ , v)      //accept Eigen pair
                 else continue
               }
             end for                        //Closing loop
  Step (6):  accept (λ , x )= (θ , v)       // finding Eigen pairs if step (5) true
}
```

Inverse Power Method ( ):

```
{
      Step (1):  read A                     // read the elements of matrix A
                 set  x = z0                //∋ ||z0|| = 1 the initial guess
                 set  ζ= ζ0                 // relative tolerant error
                 set  Imax                  // maximum  number of iteration
                 for k = 1,2, ……, Imax  do  // Starting loop
                     {
      Step (2):      Ay = x                 // Solve for y i.e. y=A^-1 x
                                            // by LU decomposition, as Wielandt
      Step (3):      x = y / ||y||2         // Normalization ,avaid over
```

```
                                        //and under flow
        Step (4):     v = Ax            //Matrix multiplication
        Step (5):     θ = x * v         //Guess Eigen value
        Step (6):     if ||y – v||2 ≤ ζ  //Stopping criterion
                      set (λ , x )= (θ , x)   //accept Eigen pair  for A^-1
                      else continue
                    }
                     end for            //Closing loop
        Step (7):    accept (λ , x ) = (1/θ , x)   //Finding Eigen pairs  for A
}
```

4.3 Proposed Modified Hybrid Iterative Algorithm

In previous section we have briefly presented the existing Power method and Inverse Power method in which the pseudo-code of the Power method is denoted as **Power Method ( )** and Inverse Power method is denoted as **Inverse Power Method ( )**. To find second (smallest) Eigen pair along with largest Eigen pair, at first, we have modified the Power method by the concept of shifting Eigen property (discussed in Chapter 2). But it is worthwhile to mention here that (Chapter V discuss in details) by using shifting property, the Power method, sometimes, fail to find out absolutely smallest Eigen value (ignore the negative sign) because of the nature of the Eigen spectrum. In such condition, Inverse Power method is necessary to find out the smallest Eigen pair. In order to find out largest as well as smallest Eigen pairs and the nature of the Eigen spectrum, we have made a hybrid algorithm by joining Power method and Inverse Power method with some modifications. The proposed **Modified Hybrid Iterative Algorithm** is given bellow:

Modified Hybrid Iterative Algorithm ( )

```
{
   Step (1):   read A                          //read the elements of matrix A
               Set B=A                         //for finding Eigen pairs of B
                                               //rather than A
               Set {λ, x}={λ0, x0}             //guess initial Eigen pairs
               for r =1,2   do                 //starting loop  for modified
                                                Power method
               {
                  if   r=1
                  {
```

Step (2): apply **Power Method** ( ) // with arguments **B(=A)**, λ_0, $\mathbf{x_0}$
output $\{\lambda_1, \mathbf{x_1}\}$ //largest Eigen pair of **B**
Step (3) : find s_1, such that $\lambda_1 = s_1|\lambda_1|$ // $s \in \{+, -\}$, sign of the Eigen value
Step (4): output $\{\lambda_1, \mathbf{x}_1, s_1\}$ //obtain largest Eigen pairs of **A**
$r = r+1$ //with sign of Eigen value
}
else if r =2
{
Step (5): set **B**=**A**- λ_1**I** //using shifting property
Step (6): apply **Power Method** ( ) //with arguments **B(≠A)**, λ_0, $\mathbf{x_0}$
output $\{\sigma_2, \mathbf{y_2}\}$ //largest Eigen pair of **B**
Step (7) : $\lambda_2 = \sigma_2 + \lambda_1$
find s_2, such that $\lambda_2 = s_2|\lambda_2|$ //$s_1 \in \{+, -\}$, sign of Eigen value
Step (8): output $\{\lambda_2, \mathbf{x}_2, s_2\}$ //obtain 2nd Eigen pairs of **A**
} //with sign of Eigen value
}
end for //ending loop for
//modified Power method
Step (9) : **if** ($s_1 = s_2$ and >0)
{
Output : $\{(\lambda_1, \mathbf{x}_1), (\lambda_2, \mathbf{x}_2), (\text{all } \lambda_i \geq 0)\}$ // largest and absolutely smallest
// Eigen values,
//corresponding vectors and
// nature of all Eigen spectrum
//which +ve

Step (10) : **Stop** //as find out all information
}
else if ($s_1 = s_2$ and <0)
{
Output : $\{(\lambda_1, \mathbf{x}_1), (\lambda_2, \mathbf{x}_2), (\text{all } \lambda_i \leq 0)\}$ // largest and absolutely smallest
//Eigen values
// corresponding vectors and
//nature of all Eigen spectrum
//which - ve
Step (11) : **Stop** // as find out all information
}
else if ($s_1 \neq s_2$ and s_1 >0)
{
Output : $\{(\lambda_1, \mathbf{x}_1), (\lambda_2, \mathbf{x}_2), (\text{sign of all } \lambda_i)\}$ //positive largest and
//negative largest Eigen values,
// corresponding vectors and

// nature of Eigen spectrum
//which are +ve to - ve
Step (12) : **continue** // as smallest Eigen values
//is not find out
// need inverse power method

}
else if ($s_1 \neq s_2$ and $s_1 < 0$)
{
Output : {(λ_1, $\mathbf{x}_1$), (λ_2, $\mathbf{x}_2$),(sign of all λ_i)} // negative largest and
// positive largest Eigen values,
// corresponding vectors and
// nature of Eigen spectrum
//which are -ve to + ve
Step (13) : **continue** // as smallest Eigen values
// is not find out
// need inverse power method

}
Step (14) ***:*** Set **B =A** //for finding smallest Eigen
//pairs of **B**
Set {λ,**x**}={λ_0, $\mathbf{x_0}$} //guess initial Eigen pairs
Step (15) : **Apply Inverse Power Method ( )** //with arguments **B(=A))**,
λ_0, $\mathbf{x_0}$
output {$\lambda_3^*, \mathbf{x_3}$} //smallest Eigen pair of **A**
Step (16) : find s_3, such that $\lambda_3^* = s_3|\lambda_3^*|$ //$s_3 \in \{+, -\}$, sigen of
Eigen
value
Step (17):
Output : {(λ_1, $\mathbf{x}_1$), (λ_3^*, $\mathbf{x}_3$),(sign of all λ_i)} //largest and absolutely
//smallest Eigen values,
//corresponding vectors and
//nature of all Eigen
spectrum
Stop (18): **Stop** and **end**
} //end the algorithm

4.4 Discussion of the Proposed Algorithm

Here we have proposed **Modified Hybrid Iterative Algorithm** based on Power method and Inverse Power method respectively to find out largest and smallest Eigen values, corresponding Eigen vectors and the nature of Eigen spectrums respectively. In the pseudo-code of proposed algorithm, we have observed that there is a for loop, with index r = 1 and 2, which is dedicated with power method. The first larger loop will start with r =1. When r =1,

for the call of function **Power Method** ( ), the algorithm will be able to produce largest Eigen pair. In consequence in Step (4), the algorithm will be able to find out the sign of largest Eigen value, which is helpful for the identification of second Eigen value. After execution of Step (4), the value of *r* will be increased to 2. So, in the second iteration within this loop, the algorithm skips step (2) to (4) and as a result, the algorithm will start execution from Step (5). In Step (5) the original matrix **A** is transformed to **B** by the shifting element λ_1 such that Eigen values of **B** are Eigen values of A but shifted by λ_1 (the largest Eigen value of A). Again the algorithm calls the function **Power Method** ( ). Therefore, again the function **Power Method** ( ) produces the largest Eigen pair of **B** rather than **A.** Consequently, in step (8), the algorithm is able to find out second Eigen pairs of the given matrix **A** successfully. As the value of $r = 2$, the algorithm escape from the first major loop and eventually enter into next consequence step namely Step (9). The Step (9) is consisting of some conditional arguments. If the sign of both Eigen values are same then without execute the function **Inverse Power Method ( )** the algorithm is able to find out both absolute largest and smallest (ignore the sign) Eigen values, corresponding Eigen vectors and natures of spectrum of Eigen values. But if the sign of both Eigen values are not same, then second Eigen value produced by **Power Method** ( ) is not absolutely (ignoring sign) smallest Eigen value though smallest in magnitude. Therefore the algorithm proceeds to next steps i.e. Step (14), (15), (16), (17) and finally (18). When the algorithm executes Step (15), then the **Inverse Power Method ( )** function is run. As a result the absolute smallest Eigen value and corresponding Eigen vector along with nature of the Eigen values are found.

CHAPTER V

Some Lemma Related to the Proposed Algorithm and Experiments

5.1 Introduction

It is known that the proposed Modified Power method is developed on the basis of the fundamental properties of Eigen values and Eigen vectors of a matrix. As it is mentioned earlier that if we have problem

$$\mathbf{Ax} = \lambda\mathbf{x} \tag{5.1}$$

where **A** be $n \times n$ matrix and let $(\lambda_1, \mathbf{x_1})$, $(\lambda_2, \mathbf{x_2})$, $(\lambda_3, \mathbf{x_3})$, ……., $(\lambda_n \mathbf{x}_n)$ are the Eigen pairs of the problem of equation (5.1). So for any constant α, we have problem

$$(\mathbf{A} - \alpha\mathbf{I})\mathbf{x} = (\lambda - \alpha)\mathbf{x} \tag{5.2}$$

$$(\mathbf{A} - \alpha\mathbf{I})\mathbf{x} = \mu\mathbf{x} \tag{5.3}$$

where $\mu = \lambda - \alpha$. Then the Eigen values of the problem of equation (5.3) are

$\mu_1, \mu_2, \mu_3, \ldots, \mu_n$ respectively. Hence Eigen pairs of equation (5.2) are $(\lambda_1 - a, \mathbf{x}_1)$, $(\lambda_2 - \alpha, \mathbf{x_2})$, $(\lambda_3 - \alpha, \mathbf{x_3})$, ……., $(\lambda_n - \alpha, \mathbf{x_n})$ respectively.

5.2 Proposed Lemma

According to the proposed algorithm, the first part of the proposed algorithm is able to find out the largest (in magnitude) Eigen value (say first value) as well as one other Eigen value (say second Eigen value) and corresponding Eigen vectors. What is the nature of all values? What will be the nature of the second value obtained by the proposed algorithm? In this regard, we have proposed four Lemma as follows.

Lemma--1: If first (largest) Eigen value is positive and second Eigen value is also positive (produced by shifting largest one) then the second value is the smallest. In consequence, all Eigen values are positive.

Lemma--2: If first (largest) Eigen value is positive and second Eigen value is negative (produced by shifting largest one) then the second Eigen value obtained by the algorithm is smallest Eigen value as well as the largest (in magnitude) negative among all negative values (if any). In consequence, some Eigen values along with largest Eigen value are positive and some Eigen values are negative.

Lemma--3: If first (largest) Eigen value is negative and second Eigen value is also negative (produced by shifting largest one) then the second Eigen value is the smallest. In consequence, all Eigen values are negative.

Lemma--4: If first (largest in magnitude) Eigen value is negative and second Eigen value is positive (produced by shifting largest one) then the second Eigen value is the largest positive Eigen value among all positive Eigen values (if any), but not necessarily be the smallest Eigen value regarding magnitude among all Eigen values. In consequence, some Eigen values along with largest Eigen values are negative in sign and some Eigen values are positive. In the following subsection, the proofs of the lemma are discussed with illustrations.

5.3 Proofs of these Lemma and illustrations

Proof of Lemma--1: Let us consider the equation (5.1) and (5.2) and there corresponding pairs. To proof the Lemma 1, we consider all the Eigen values of equation (5.1) are positive and say $\lambda_1 > \lambda_2 > \lambda_3 > \cdots \lambda_n \geq 0$. As the largest Eigen value as well as all Eigen values are positive. Therefore we must have $|\lambda_1| > |\lambda_i| : i = 2,3,\cdots,n$. So first Eigen value obtained by the algorithm is largest Eigen value and it is obviously λ_1.

To find out the second Eigen value, the proposed algorithm will consider the equation (5.2) with $\alpha = \lambda_1$. So the corresponding Eigen pairs of equation (5.2) are $(\lambda_1 - \lambda_1, \mathbf{x}_1)$, $(\lambda_2 - \lambda_1, \mathbf{x}_2)$, $(\lambda_3 - \lambda_1, \mathbf{x}_3), \cdots$, $(\lambda_n - \lambda_1, \mathbf{x}_n)$ respectively. Since λ_1 is the largest Eigen value (with positive magnitude), so all Eigen values of equation (5.2) must be now non-positive i.e. $(\lambda_1 - \lambda_1)(= 0) > (\lambda_2 - \lambda_1) > (\lambda_3 - \lambda_1) > \cdots > (\lambda_n - \lambda_1)$. So by taking absolute Eigen value of each term, we have $|\lambda_n - \lambda_1| > |\lambda_i - \lambda_1| : i = 2,3,\cdots,n$. So, in second iteration, the proposed algorithm will find the Eigen value of the equation (5.2) which is must be $\mu_1 = \lambda_n - \lambda_1$. Therefore $\lambda_n = \mu_1 + \lambda_1$, which is the smallest value of equation (5.1). Hence proved the Lemma – 1. Now for the illustration of the Lemma – 1, we will consider the following example 1.

Example 1: $\mathbf{A} = \begin{bmatrix} 4 & 3 & 1 & 0 \\ 3 & 4 & 2 & 1 \\ 1 & 2 & 7 & 1 \\ 1 & 1 & 1 & 8 \end{bmatrix}$

Solution: By using Mat Lab solver, we have the values of the given problem: 10.1375, 6.9741, 4.9759, 0.9125. So by comparing with equation (5.1) we have $\lambda_1(= 10.1375) > \lambda_2(= 6.9741) > \lambda_3(= 4.9759) > \lambda_4(= 0.9125) > 0$. So our proposed algorithms, in first iteration, is able to find the largest value $\lambda_1(= 10.1375)$. In second iteration, the algorithm will find the largest value of equation (5.2) where the value of α = 10.1375. Therefore values of equation (5.2) must be μ = 0, -3.1634, -5.1616, -9.2250. Therefore, in second iteration, the proposed algorithm is able to find the largest (in magnitude) value of equation (5.2) which is obviously μ_1 = -9.2250. Finally the algorithm finds the second value of equation (5.1) as

$\lambda = \lambda_1 + \mu_1 = 10.1375 + (-9.2250) = 0.9125$, which is the smallest value of the given problem.

Proof of Lemma--2: Let us consider the equation (5.1) and (5.2) and their corresponding pairs. To proof the Lemma 2, let us consider the largest value λ_1 of equation (5.1) is positive and among the remain values, some of them are positive and some of them are negative in sign; say $\lambda_1 > \lambda_2 > \lambda_3 > \lambda_m > 0 > \lambda_{m+1} > \lambda_{m+2} > \cdots > \lambda_n$. Here λ_n be the smallest value (but not necessarily smallest in magnitude). As the largest value is positive so we must have $|\lambda_1| > |\lambda_i| : i = 2,3,\cdots,n$. So first value obtained by the algorithm is largest value and it is obviously λ_1.

To find the second value, the proposed algorithm will consider the equation (5.2) with $\alpha = \lambda_1$. So the corresponding pairs of equation (5.2) become $(\lambda_1 - \lambda_1, \mathbf{x}_1)$, $(\lambda_2 - \lambda_1, \mathbf{x_2})$, $(\lambda_3 - \lambda_1, \mathbf{x_3}), \cdots, (\lambda_n - \lambda_1, \mathbf{x}_n)$ respectively. Since λ_1 is the largest value, so all values of equation (5.2) must be non-positive i.e.

$(\lambda_1 - \lambda_1)(= 0) \geq (\lambda_2 - \lambda_1) > (\lambda_3 - \lambda_1) > \cdots > (\lambda_m - \lambda_1) > (\lambda_{m+1} - \lambda_1) > (\lambda_{m+2} - \lambda_1) > \cdots > (\lambda_n - \lambda_1)$. So by taking absolute value of each term, we have $|\lambda_n - \lambda_1| > |\lambda_i - \lambda_1| : i = 2,3,\cdots,n$. Therefore, in second iteration, the proposed algorithm will able to find the value of the equation (5.2) which is must be $\mu_1 = \lambda_n - \lambda_1$. Therefore $\lambda_n = \mu_1 + \lambda_1$, which is the smallest Eigen value of equation (5.1) by considering sign of each value. Moreover this the largest (in magnitude) value among the all negative values. Hence proved the Lemma – 2. Now for the illustration of the Lemma – 2, we have considered the following example 2.

Example 2: $\mathbf{B} = \begin{bmatrix} 2 & 2 & 1 & 2 \\ 8 & 2 & 1 & 4 \\ 9 & 2 & 2 & 5 \\ 1 & 2 & 5 & 6 \end{bmatrix}$

Solution: By using Mat Lab solver, we have the values of the given problem are: 12.7568, 1.4718, 0.4045, −2.6332.. So by comparing with equation (5.1) we have $\lambda_1(= 12.7568) > \lambda_2(= 1.4718) > \lambda_3(= 0.4045) > 0$ and $\lambda_4(= -2.6332) < 0$. Here we have observed the largest value $\lambda_1(= 12.7568)$ is positive. So, in first iteration, our proposed algorithms is able to find the largest value $\lambda_1(= 12.7568)$. In second iteration the algorithm is find the largest

value of equation (5.2) where value of α = 12.7568. Therefore values of equation (5.2) must be μ = 0, -11.2850, -12.3523, -15.3900, all values are non-positive. Therefore, in second iteration, the proposed algorithm is able to find the largest (in magnitude) value of equation (5.2) which is obviously $\mu_1 = -15.3900$. Finally the algorithm finds the second value of equation (5.1) as $\lambda = \lambda_1 + \mu_1 = 12.7568 + (-15.3900) =$ -2.6332 which is smallest value among the all value but it is the largest (in magnitude) value among the all negative values $\{-2.6332\}$ of the given problem.

Proof of Lemma--3: Let us consider the equation (5.1) and (5.2) and their corresponding pairs. To proof the Lemma 3, we consider all the values of equation (5.1) are negative and say $\lambda_1 < \lambda_2 < \lambda_3 < \cdots < \lambda_n < 0$. Therefore we must have $|\lambda_1| > |\lambda_i| : i = 2,3,\cdots,n$. So first value obtained by the algorithm is largest (in magnitude) value and it is obviouslyλ_1.

To find the second value, the proposed algorithm will consider the equation (5.2) with $\alpha = \lambda_1$. So the corresponding pairs of equation (5.2) are $(\lambda_1 - \lambda_1,\ X_1)$, $(\lambda_2 - \lambda_1,\ X_2)$, $(\lambda_3 - \lambda_1,\ X_3), \cdots, (\lambda_n - \lambda_1,\ X_n)$ respectively. Since λ_1 is the largest (in magnitude) but negative in sign value i.e. $\lambda_1 = -|\lambda_1|$, so all values of equation (5.2) must non-negative i.e. $(\lambda_1 - \lambda_1)(= 0) < (\lambda_2 - \lambda_1) < (\lambda_3 - \lambda_1) < \cdots < (\lambda_n - \lambda_1)$. So by taking absolute value of each term, we have $|\lambda_n - \lambda_1| > |\lambda_i - \lambda_1| : i = 2,3,\cdots,n$. Therefore, in second iteration, the proposed algorithm will able to find the value of the equation (5.2) which is must be $\mu_1 = \lambda_n - \lambda_1$. Therefore $\lambda_n = \mu_1 + \lambda_1$, which is the smallest (in magnitude) value of equation (5.1). Hence proved the Lemma – 3. Now for the illustration of the Lemma – 3, we will consider the following example 3.

Example 3 : $\mathbf{C} = \begin{bmatrix} -26 & -26 & -17 & -4 \\ -27 & -30 & -26 & -14 \\ -18 & -26 & -55 & -17 \\ -16 & -17 & -18 & -66 \end{bmatrix}$

Solution: By using Mat Lab solver, we have the values of the given problem are: -102.7695, -48.6385, -24.7594, -0.8326. So by comparing with equation (5.1) we have $\lambda_1(= -102.7695) < \lambda_2(= -48.6385) < \lambda_3(= -24.7594) < \lambda_4(= -0.8326) <0$. So in first iteration, our proposed algorithms is able to find the largest (in magnitude) value $\lambda_1(= -102.7695)$. In second iteration, the algorithm will be find the largest value of equation (5.2)

where value of $a = -102.7695$. Therefore values of equation (5.2) must be μ = 0, 54.1310, 78.0101, 101.9369. Therefore, in second iteration, the proposed algorithm is able to find the largest value of equation (5.2) which is obviously μ_1 = 101.9369. Finally the algorithm find the second value of equation (5.1) as $\lambda = \lambda_1 + \mu_1 = -102.7695 + 101.9369 = -0.8326$, which is the smallest Eigen value of the given problem.

Proof of Lemma--4: Let us consider the equation (5.1) and (5.2) and there corresponding pairs. To proof the Lemma 4, let us consider the largest (in magnitude) value λ_1 of equation (5.1), is negative and among the remain values, some of them are positive and some of them are negative in sign; say $\lambda_1 < \lambda_2 < \lambda_3 < \cdots < \lambda_m < 0$ and $\lambda_{m+1} > \lambda_{m+2} .> \cdots > \lambda_n > 0$. As λ_1 be the largest value regarding magnitude, so we must have $|\lambda_1| > |\lambda_i| : i = 2,3,\cdots,n$. So the first value obtained by the algorithm is largest (in magnitude) value and it is obviously λ_1.

To find the second value the proposed algorithm will consider the equation (5.2) with $\alpha = \lambda_1$. So The corresponding pairs of equation (5.2) are $(\lambda_1 - \lambda_1, X_1)$, $(\lambda_2 - \lambda_1, X_2)$, $(\lambda_3 - \lambda_1, X_3), \cdots, (\lambda_n - \lambda_1, X_n)$ respectively. Since λ_1 is the largest but negative in sign i.e. $\lambda_1 = -|\lambda_1|$ among the all values of equation (5.1) , so all values of equation (5.) must be non-negative and consequence $(\lambda_1 - \lambda_1)(= 0) < (\lambda_2 - \lambda_1) < (\lambda_3 - \lambda_1) < \cdots < (\lambda_m - \lambda_1) < (\lambda_{m+1} - \lambda_1) < (\lambda_{m+2} - \lambda_1) < \cdots < (\lambda_n - \lambda_1)$. So by taking absolute value of each term, we must have $|\lambda_n - \lambda_1| > |\lambda_i - \lambda_1| : i = 2,3,\cdots,n$. Therefore, in second iteration, the proposed algorithm will able to find the value of the equation (5.2) which is must be $\mu_1 = \lambda_n - \lambda_1$. Therefore $\lambda_n = \mu_1 + \lambda_1$which is the largest positive value among all positive values of equation (5.1) if any. Hence proved the Lemma – 4. Now for the illustration of the Lemma – 4, we will consider the following example 4.

Example 4 : $\mathbf{D} = \begin{bmatrix} -6 & 2 & 1 & -2 \\ 8 & 2 & 1 & 4 \\ 9 & 2 & 2 & 5 \\ 1 & 7 & 5 & -6 \end{bmatrix}$

Solution: By using Mat Lab solver, we have the values of the given problem are: −12.4503, − 4.1452, 0.6327, 7.9628. So by comparing with equation (5.1) we have $\lambda_1(= -12.4503) <$

$\lambda_2(=-4.1452) < 0 < \lambda_3(=0.6327) < \lambda_4(=7.9628)$. Therefore $|\lambda_1(=-12.4503)| > |\lambda_i| : i = 2,3,4$. So, in first iteration, our proposed algorithms is able to find the largest (in magnitude) value $\lambda_1(=-12.4503)$. In second iteration, the algorithm is able to find the largest (in magnitude) value of equation (5.2) where value of $a = -12.4503$. Therefore values of equation (5.2) must be μ = 0, 8.3051, 13.0830, 20.4131. Therefore, in second iteration, the proposed algorithm is able to find the largest value of equation (5.2) which is obviously μ_1 = 20.4131. Finally the algorithm find the second value of equation (5.1) as $\lambda = \lambda_1 + \mu_1 = -12.4503 + (20.4131) = 7.9628$, which is the largest positive value among the all positive values {0.6327, 7.9628} of the given problem.

5.4 Experimental Results

In previous section some Lemma as well as illustration are proved. Now we have performed some experiments on the proposed algorithm to justify the proposed lemma regarding the proposed modified hybrid algorithm. In this regards, we have considered the Eigen value problem of the form equation (5.1). For the first experiment, we have considered the transformation operator **A** is given as bellow:

Test Prob. 1

$$\mathbf{E} = \begin{bmatrix} 1 & 2 & 4 & 5 & 6 & 7 & 8 & 3 & 5 & 8 & 1 & 2 \\ 2 & 2 & 9 & 8 & 5 & 6 & 5 & 2 & 1 & 0 & 2 & 0 \\ 4 & 9 & 1 & 2 & 9 & 5 & 6 & 5 & 0 & 2 & 1 & 2 \\ 5 & 8 & 2 & 9 & 5 & 2 & 1 & 0 & 2 & 1 & 3 & 4 \\ 6 & 5 & 9 & 5 & 5 & 2 & 1 & 2 & 0 & 5 & 8 & 9 \\ 7 & 6 & 5 & 2 & 2 & 1 & 9 & 5 & 6 & 2 & 1 & 4 \\ 8 & 5 & 6 & 1 & 1 & 9 & 1 & 2 & 3 & 4 & 2 & 5 \\ 3 & 2 & 5 & 0 & 2 & 5 & 2 & 2 & 0 & 1 & 0 & 2 \\ 5 & 1 & 0 & 2 & 0 & 6 & 3 & 0 & 1 & 2 & 5 & 9 \\ 8 & 0 & 2 & 1 & 5 & 2 & 4 & 1 & 2 & 6 & 2 & 2 \\ 1 & 2 & 1 & 3 & 8 & 1 & 2 & 0 & 5 & 2 & 1 & 0 \\ 2 & 0 & 2 & 4 & 9 & 4 & 5 & 2 & 9 & 2 & 0 & 5 \end{bmatrix}$$

Table 5.1 Finding Eigen pairs and comparison of values for the **Test Prob. 1**

Eigen pairs and sign of Eigen value obtained by Modified Hybrid Iterative Algorithm				Eigen values (Mat Lab).
Eigen value	Sign	Eigen values (λ)	Eigen vector (**X**)	
First Eigen Value (Largest Eigen value) (in magnitude)	+ *ve*	*43.6996*	[0.92073, 0.782303, 0.851719, 0.767980, 1.000000, 0.876090 , 0.849523, 0.453403, 0.599790, 0.641932, 0.480074, 0.799947]	*43.6996* *12.1565*

Second Eigen value (second Largest negative Eigen value (among the negative values)	- *ve*	*-15.8665*	[-0.708442,-0.233479,-0.394794, 0.193943, 1.00000, -0.124613, 0.721741, 0.195695, 0.801014, 0.0019818, -0.731598, -0.876232]	*10.8018* *7.8891* *5.9091* *1.8160* *-0.7210* *-4.9082* *-6.5833* *-8.4682* *- 0.7250* *- 15.8666*
Smallest Eigen value (in magnitude)	- *ve*	*-0.72098*	[0.10929, -0.261378, -0.274267, 0.241883, -0.106012 ,0.0853739, -0.394766, 1.000000, 0.174666, -0.0421989, 0.421505, -0.26894]	
Second smallest positive Eigen value	+ *ve*	*1.815937*	[-0.115199 ,0.224341 , -0.00357368, -0.416398 , -0.0414743 ,0.189909, 0.21043 , -0.408524 ,0.382283, -0.228485 ,1 ,-0.535562]	

It is observed that for the **Test Prob. 1** the transformation matrix **E** is of order 12. The experimental results are displayed in the Table 5.1. For the validity of the proposed algorithm, the problem is also solved by Mat Lab solver. The Eigen values obtained by the solver is also displayed in the Table 5.1 namely last column of the table. At first the proposed algorithm find out the first value which is *43.6996* and sign of the value is positive. Now we observe that the largest value obtained by the Mat Lab. solver is also 43.6996. That is the algorithm is successfully able to find out the largest value. Now we observe that the second value obtained by the algorithm is *-15.*8665 and sign of the second value is negative. So according to the lemma 2, the second value is not smallest (in magnitude) value but largest value among all the negative values. Moreover the spectrums of value consist of both positive and negative values in which the largest value is of positive sign. We observe that the experimental results agree with the lemma. That is the second value is not smallest value regarding magnitude but largest in magnitude among all negative value though smallest value if we consider sign of each value. As the second value is not absolutely smallest value, so to find out the smallest (in magnitude) values and corresponding vector, the algorithm has proceed further steps and consequence algorithm has executed **Inverse Power method ( )** function. As a result the algorithm is able to find out the value which value is *-0.72098* and sign is negative**.** It is notice that the smallest value obtained by the Mat Lab is *-0.7210* which is almost identical with the value obtained by the proposed algorithm. Therefore from this experiment we may

conclude that the proposed algorithm successfully able to find out both absolutely largest as well as absolutely smallest values and the nature of the spectrums. We also observe that the algorithm able to find not only values but also able to find corresponding vectors which is displayed in the column 4 of the Table 5.1.

Now for verification of the validity of the values and corresponding vectors, we have considered the equation 5.1. In that equation, we put **A**=**E** and also put the value of each **x** given in the Table 5.1. We have solved left hand side of the equation 5.1 for each value of **x**. Now for each pair (λ, **x**) we compute **Ax**/ λ. The computational results are shown in the Table 5.2. It is notice that, each pair (λ, **x**) satisfy the equation 5.1. This implies that the Eigen pairs obtained by the proposed algorithm are valid solutions.

Table 5.2: Verification of the Eigen pairs obtained by the proposed algorithm for the **Test Prob. 1**

A	**X**	**AX**	=	λ**X**		λ	**X**(=λ**X**/ λ)
E	0.9207	40.2355		40.2355		*43.6996*	0.9207
	0.7823	34.1863		34.1863			0.7823
	0.8517	37.2198		37.2198			0.8517
	0.7680	33.5606		33.5606			0.7680
	1.0000	43.6996		43.6996			1.0000
	0.8761	38.2848		38.2848			0.8761
	0.8495	37.1238		37.1238			0.8495
	0.4534	19.8135		19.8135			0.4534
	0.5998	26.2106		26.2106			0.5998
	0.6419	28.0522		28.0522			0.6419
	0.4801	20.9790		20.9790			0.4801
	0.7999	34.9573		34.9573			0.7999
E	-0.7084	11.2407		-0.7085		*-15.8666*	-0.7084
	-0.2335	3.7048		-0.2335			-0.2335
	-0.3948	6.2638		-0.3948			-0.3948

	0.1939	-3.0773		0.1940			0.1939
	1.0000	-15.8665		1.0000			1.0000
	-0.1246	1.9770		-0.1246			-0.1246
	0.7217	-11.4515		0.7217			0.7217
	0.1957	-3.1049		0.1957			0.1957
	0.8010	-12.7094		0.8010			0.8010
	0.0020	-0.0315		0.0020			0.0020
	-0.7316	11.6079		-0.7316			-0.7316
	-0.8762	13.9029		-0.8762			-0.8762
E	0.1093	-0.0783		-0.0783			0.1086
	-0.2614	0.1885		0.1885			-0.2615
	-0.2743	0.1977		0.1977			-0.2742
	0.2419	-0.1743		-0.1743			0.2418
	-0.1060	0.0764		0.0764			-0.1060
	0.0854	-0.0610		-0.0610		*-0.7210*	0.0845
	-0.3948	0.2849		0.2849			-0.3951
	1.0000	-0.7210		-0.7210			1.0000
	0.1747	-0.1258		-0.1258			0.1745
	-0.0422	0.0306		0.0306			-0.0425
	0.4215	-0.3039		-0.3039			0.4215
	-0.2689	0.1940		0.1940			-0.2691

Now for further experimental study, we have considered the Test Prob. 2 given bellow. We observe that in the **Test Prob. 2**, the transformation matrix **F** is of order 8. To study the performance of the proposed algorithm, we have, at first, implement the proposed algorithm to find out absolutely largest and smallest values and corresponding vectors. The experimental outputs are displayed in the Table 5.3. For the comparison study, we have find out all the values of the problem by Mat lab solver. These values are also shown in the column 5 of the Table 5.3.

Test Prob. 2

$$\mathbf{F} = \begin{bmatrix} 57 & 37 & 50 & 21 & 25 & 22 & -3 & 20 \\ 37 & 44 & 40 & 28 & 12 & 29 & 27 & 07 \\ 50 & 40 & 85 & 31 & 40 & 36 & 23 & 22 \\ 21 & 28 & 31 & 80 & 7 & 18 & 16 & 15 \\ 25 & 12 & 40 & 7 & 59 & 20 & 15 & 08 \\ 22 & 29 & 36 & 18 & 20 & 34 & 24 & 10 \\ -3 & 27 & 23 & 16 & 15 & 24 & 11 & -3 \\ 20 & 07 & 0\,2 & 15 & 8 & 10 & -3 & 61 \end{bmatrix}$$

We notice that the first value obtained by the algorithm is *229.048* which is positive in sign. It is observed in the table that the first value obtained by the algorithm is almost identical with largest value obtained by the Mat Lab solver. Again we observe that the second value obtained by the proposed algorithm is *2.4725* with positive sign. Now according to the Lemma 1 the second value must be smallest value and consequence all values must be positive in sign. We observe that the second value is almost identical both in magnitude as well as sign with the smallest value obtained by the Mat Lab solver. It is shown in the table that all the values of the problem obtained by the Mat Lab. solver are positive. That is the experimental results agree with the proposed Lemma.

Table 5.3 Finding Eigen pairs and comparison of Eigen values for the **Test Prob. 2**

Eigen pairs and sign of Eigen value obtained by Modified Hybrid Iterative Algorithm				Mat Lab
Eigen value	Eigen values (λ)	Sign	Eigen vector (**X**)	All Eigen values
1st Eigen value (Largest Eigen value)	*229.048*	+ve	[0.905122,0.789705,0.877491,0.733968,1, 646421,0.474731,0.91332,0.668277]	*229.0477* 108.0800 68.7215 50.3789 38.0092 18.0295
2nd Eigen value	*2.4725*	+ve	[-0.714502, 1, 0.109971, -0.100363, ,0.257215 ,-0.540873 ,-0.191723, 0.156506]	

(Smallest Eigen value)				15.2773 *2.4559*

Again for the verification of the validity of the values and corresponding vectors, we have consider the equation 5.1. In that equation, now we put **A**=**F** and also put the value of each **x** given in the Table 5.4. We have solved left hand side of the equation 5.1 for each Eigen value of **x**. Now for each pair (λ, **x**) we compute **Ax**/ λ. The computational results are shown in the Table 5.4. It is notice that, each pair (λ, **x**) satisfy the equation 5.1. This implies that the Eigen pairs obtained by the proposed algorithm are valid solutions.

Table 5.4: Verification of the Eigen pairs obtained by the proposed algorithm for the **Test Prob.**

A	**X**	**AX**	=	**λX**		λ	**X**(=**λX**/ λ)
F	0.6926	158.6375		158.6375		229.048	0.6926
	0.6715	153.8034		153.8034			0.6715
	1.0000	229.0477		229.0477			1.0000
	0.6233	142.7703		142.7703			0.6233
	0.5446	124.7358		124.7358			0.5446
	0.5657	129.5751		129.5751			0.5657
	0.5858	134.1667		134.1667			0.5858
	0.3461	79.2698		79.2698			0.3461
F	-0.7145	-2.0992		-2.0992		2.4725	-0.7145
	1.0000	2.4724		2.4724			1.0000
	0.1100	0.3619		0.3619			0.1100

	-0.1004	-0.2797		-0.2797			-0.1007
	0.2572	0.5682		0.5682			0.2572
	-0.5409	-0.8483		-0.8483			-0.5409
	-0.1917	-0.6147		-0.6147			-0.1918
	0.1565	0.3949		0.3949			0.1566

For the above experimental study we may conclude that the proposed Modified Hybrid Iterative Algorithm able to find absolutely largest as well as absolutely smallest values and corresponding vectors successfully. Moreover the experimental results agree with the proposed Lemma in perspective to the algorithm.

Now to analysis the performance of the proposed algorithm, we have considered few algebraic examples with several dimensions. At first, let us consider an algebraic problem of the form:

$$\mathbf{Ax} = \lambda \mathbf{x}$$

Now we will consider some examples to find out Eigen pairs by the proposed algorithm. At first we consider a 4×4 matrix A, where

Test Prob. 3: $$\boldsymbol{G} = \begin{bmatrix} -3 & -2 & -2 & 4 \\ -2 & -3 & -2 & 1 \\ -1 & -1 & -2 & 1 \\ -2 & -2 & -2 & 1 \end{bmatrix}$$

Whose Eigen values are $\lambda = \{-3, -2, -1, -1\}$ obtained by some package software.

Table 5.5 The experimental results for **Test Prob. 3**, in which all Eigen values are negative.

	for r =1 (find first Eigen pair)	for r =2 (find second Eigen pair)
No. of Iteration	37	28
Eigen Vector	[1.94267e-007 ,1 ,0.333333 ,0.666667]	[-0.9655, 1.0000, -0.0345, -0.0000]
Eigen Value	**-3.00**	**-1.00**

Ax = **λx**	[0.0000, -3.0000, -1.0000, -2.0000] = -3×[1.94267e-007 ,1 ,0.333333 ,0.666667]	[0.9655, -1.0000, 0.0345, -0.0000] = -1×[-0.9655, 1.0000, - 0.0345, -0.000]

Now we have performed experiment on this matrix **G** by our proposed algorithm to find the Eigen pairs. The experimental results are given in Table 5.5. It is observed in the table that, for the first loop of the algorithm i.e. r =1 the proposed algorithms is successfully able to find out largest in magnitude Eigen value $\lambda = -3$ and corresponding Eigen vector. Similarly for r = 2, the algorithm is able to find out the second Eigen value $\lambda = -1$. As we know in this problem all Eigen values are negative i.e. $\lambda = \{-3, -2, -1, -1\}$ so this second Eigen value is obviously smallest Eigen value. Consequence, the algorithm need not execute the function **Inverse Power method ( ).** Therefore safe much computation cost as well as numerical complexity. It is also notice in the table that the verification of the Eigen values and corresponding Eigen vectors is shown in the Table 5.5. It is also observed in the table since the ratio of 3/2 is not very large than one so algorithm required a bit more iterations to converse its spectrum radius. Similarly the ratio of (-1+4) / (-2+4) is one so though the order of matrix is small but the algorithm need more iteration to converge the smallest Eigen pairs. It is notice that there exists algebraic multiplicity two at the characteristic root -1.

Now we have performed another experiments by considering the matrix 8×8 matrix A, which is given in **Test Prob. 4**.

Test Prob. 4: $$H = \begin{bmatrix} 12.62 & -13.60 & -2.68 & 1.22 & -3.47 & 6.14 & 3.14 & -2.30 \\ -13.60 & 22.06 & 0.90 & -2.29 & 5.24 & -11.02 & -3.87 & 3.10 \\ -2.68 & 0.90 & 3.93 & -0.48 & -0.81 & -2.10 & -0.48 & -0.10 \\ 1.22 & -2.29 & -0.48 & 1.79 & -0.21 & 0.80 & 0.27 & -0.50 \\ -3.47 & 5.24 & -0.81 & -0.21 & 3.85 & -3.04 & -1.02 & 0.82 \\ 6.14 & -11.02 & -2.10 & 0.80 & -3.04 & 11.74 & 1.00 & -1.67 \\ 3.14 & -3.87 & -0.48 & 0.27 & -1.02 & 1.00 & 1.91 & -0.41 \\ -2.30 & 3.10 & -0.10 & -0.50 & 0.82 & -1.67 & -0.41 & 2.34 \end{bmatrix}$$

Whose Eigen values are $\lambda = \{40.7214, 6.5459, 5.5471, 2.6292, 1.9870, 1.4547, 0.9258, 0.4371\}$ obtained by some package software.

Table 5.6: The experimental result for **Test Prob. 4** in which all Eigen values are positive

	for r =1	for r =2
Iteration	**13**	216
Eigen vector	[-0.690437, 1, 0.105328, -0.0983584, 0.261668, -0.579705, -0.181056, 0.155929]	[0.687959 ,0.672025,1,0.629365,0.543817 ,0.567436 ,0.595655 ,0.346538]
Eigen value	*40.7214*	*0.4371*
Ax = λx	[-28.1155, 40.7214, 4.2891, -4.0052, 106554, -23.6063, -7.3728, 6.3496] = *40.7214*×[-0.690437, 1, 0.105328, -0.0983584 ,0.261668 ,-0.579705 , -0.181056 ,0.155929]	[0.3007, 0.2937, 0.4371, 0.2751, 0.2377, 0.2480, 0.2603, 0.1514] = *0.4371*×[0.687959 ,0.672025 ,1 ,0.629365 ,0.543817 ,0.567436 ,0.595655 ,0.346538]

Now we have performed experiment on this matrix **H** by our proposed algorithm to find the Eigen pairs. The experimental results are given in Table 5.6. It is observed in the table that the proposed algorithms successfully is able to find out largest Eigen value and corresponding Eigen vector if the loop of the algorithm for r =1. Similarly for r = 2, the algorithm is able to find out the second Eigen pairs. As we know, some Eigen values are positive and some Eigen values are negative, so this second Eigen value is obviously smallest Eigen value. Consequence, the algorithm need not execute the function **Inverse Power method ( ).** Therefore safe much computation cost as well as numerical complexity. It is also notice in the table that showed the verification of the Eigen values and corresponding Eigen vectors. It is note that since the ratio of 40.7214/6.5459 are large, so algorithm requires few iteration (here 13) to convergence to demonian Eigen value. On the other hand as the ration of (0.9258-40.7214) /(1.4547-40.7214) is near to one, so algorithm need much effort (need 216 iterations) to converge the smallest Eigen value and corresponding Eigen vectors.

CHAPTER VI

Discussion and conclusion

In this thesis we have considered both power method and inverse power method as a base to develop an algorithm. It is notice that if all the Eigen values are either positive are negative, then simple shifting property on power method is enough to find out the both (largest as well as smallest) Eigen pairs, instead of calculation of inverse of the matrix with is computationally expensive as well as unstable. We have proposed a modified hybrid iterative method by hybridized both Power method as well as inverse power method along with shifting property on part of Power method rather than inverse Power method. The purpose of the proposed algorithm is to find out both largest and smallest Eigen values and corresponding Eigen vectors along with the nature of sign of the spectrum. In some physical system, there may need both largest and smallest Eigen pairs, this algorithm efficiently able to serve the purpose. Furthermore we have proposed four lemma regarding the nature of sign of Eigen values regarding the proposed algorithm.

Intensive experiments have been performed for the test of validity of the proposed algorithms as well as the lemma. The experimental results agree with proposed algorithm as well as proposed lemma. The proposed algorithm able to find out the largest Eigen- pairs as well as

smallest Eigen pairs efficiently. Moreover algorithm is able to find out the sign of spectrum. Besides the experimental study regarding lemma, we have also theoretically proof the lemma.

As the proposed algorithm has developed based on Power method, the proposed algorithm preserves some limitations as Power method. Though proposed algorithm able to find out more than one Eigen pair namely largest and smallest Eigen pair whereas Power method find only largest Eigen pair only, the proposed algorithm does not able to find out largest (in magnitude) complex Eigen value and corresponding Eigen pairs directly. The rate of convergence is linear as like power method and the speed of rate of convergence is slow if the ratio of largest Eigen value and second largest Eigen value is near or equal to one. Similar to Power method, the proposed algorithm is efficient in the cases of sparse large matrix, positive (Semi-positive) matrix.

In future, the proposed algorithm will be extended to find out other necessary Eigen pairs. Moreover the proposed algorithm will be modified by taking some theoretical concepts to overcome the limitations of the existing algorithm.

REFERENCES

1 Bai Z., (1995), Progress in the numerical solution of the non symmetric Eigen value problem, Numer. Linear Algebra Appl. pp. 219-234.

2 Bai Z., D. Day and Q.Ye, (1995), ABLE: An adaptive block Lanczos method for non-Hermitian Eigen value problems, Technical Report Research Report 95-04, University of Kentucky, Lexington, KY.

3 Bai Z., J. Demmel, J. Dongarra A. Ruhe and H. van der Vorst, (1999), Templates for the Solution of Algebraic Eigen value Problems: a Practical Guide, SIAM, Philadelphia, PA.

4 Booth T. E., (2003), Nucl. Sci. Eng. 143, pp. 291.

5 Booth T. E., (2006) Nucl. Sci. Eng. 154, pp. 48.

6 Chu M. and G. Golub, (2002), Structured Inverse Eigen value Problems, Acta Numer., Vol. 11, pp. 1-71.

7 Demmel J., (1997), Applied Numerical Linear Algebra, SIAM, Philadelphia.

8 Demmel J. and W. James, (1997), Applied Numerical Linear Algebra, Philadelphia, PA:Society for Industrial and Applied Mathematics,ISBN 0-89871-389-7, MR 1463942.

9 Demmel J.W., (1997), Applied Numerical Linear Algebra, SIAM, Philadelphia, PA.

10 Demmel J.W. and K. Veselic, (1992), Jacobi's method is more accurate than QR, SIAM J. Matrix Anal. Appl. Vol. 13, pp. 1204-1246.

11 Dyson F. J., (1962), Statistical theory of the energy levels of complex systems i, ii, and iii. Journal of Mathematical Physics, Vol. 3(1), pp.166-175.

12 Forrester P. J., N. C. Snaith, and J. Verbaarschot, (2003), Developments in random matrix theory. Journal of Physics A: Mathematical and General, 36:pp. R1-R10.

13 Gade K. K., (2009), Numerical Estimation of the Second Largest Eigen value of a Reversible Markov Transition Matrix, Doctor of Philosophy, Department of Computer Science, New York University January. An Inexact Cayley Transform Method For Inverse Eigen value Problems.

14 Golub G. H. and C. F. Van Loan, (1989), Matrix Computations, John Hopkins University Press, Baltimore, 2nd ed.

15 Golub and C. Van Loan, (1996), Matrix Computations, The Johns Hopkins University Press, Baltimore, third edition.

16 Google. (2005), http://www.google.com.

17 Gotze J., S. Paul and M. Sauer, (1993), An efficient Jacobi-like algorithm for parallel Eigen value computation, IEEE Transactions on Computers, Vol. 42, no. 9, pp. 1058 – 65, Sept.

18 Guillaumin, J. Verbeek and C. Schmid, (2009), Is that you? Metric learning approaches for face identification. In IEEE 12th International Conference on Computer Vision, pp 498–505.

19 Gubernatis J. E. and T. E. Booth, (2008), Multiple Extremal Eigen pairs by the Power Method, Theoretical Division and Applied Physics Division, Los Alamos National Laboratory, Los Alamos, NM 87545 U.S.A. T. E. Booth, LA-UR-07-4046.

20 Hochstenbach M. E. and Y. Notay, (2004), The Jacobi-Davidson method, GAMM Mitt.

21 Hoffman K. and R. Kunze, (1971), Characteristic Values §6.2 in Linear Algebra, Englewood Cliffs, NJ: Prentice-Hall, pp. 182.

22 Hoi S. C. H., W. Liu, M. R. Lyu, and W.-Y. Ma, (2006), Learning distance metrics with

contextual constraints for image retrieval. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 2072–2078.

23 Jarlebring. E., (2008), The spectrum of delay-differential equations: numerical methods, stability and perturbation PhD thesis, Inst. Comp. Math, TU Braunschweig.

24 Jian B. Z., (2004), Numerical Methods for Inverse Eigen value Problems by A Thesis Degree of Doctor of Philosophy, in Mathematics, The Chinese University of Hong Kong.

25 Karoui N. E., (2006), A rate of convergence results for the largest Eigen value of complex white wishart matrices, The Annals of Probability, Vol. 34, No. 6, pp. 2077–2117 DOI: 10.1214/009117906000000502.

26 Kato and N. Nagano, (2010), Metric learning for enzyme active-site search. Bioinformatics, Vol. 26, pp2698- 2704.

27 Kressner D., (2009), A block Newton method for nonlinear Eigen value problems.

28 Lanczos C., (1950), An iterative method for the solution of the Eigen value problem of linear differential and integral operators, Journal of Research of the National Bureau of Standards, Vol.45, pp. 255–282.

29 Lehoucq R. and K. Meerbergen, (1999), Using generalized Cayley transformations within an inexact rational Krylov sequence method, SIAM J. Matrix Anal. Appl. 20 pp:131-148.

30 Liao B.-S., Z. Bai, L.-Q. Lee and K. Ko, (2010), Nonlinear Rayleigh-Ritz iterative method for solving large scale nonlinear Eigen value problems, Taiwanese J. Math., Vol. 14(3A), pp. 869–883.

31 Marcus M. and H. Minc, (1988), Introduction to Linear Algebra, New York: Dover, pp. 145.

32 McCormick S. F. and T. Noe, (1977) Simultaneous iteration for the matrix Eigen value problem, Linear Algebra and its Applications , Vol. 16 (1), pp. 43–56.

33 McNew N., (2011), The Eigen value Gap and Mixing.

34 Mehta M. L., (1960), On the statistical properties of the level-spacing's in nuclear spectra. Nuclear Physics, Vol.18(0), pp 395-419.

35 Mehta M. L., (2004), Random Matrices, volume 142. Elsevier.

36 Mehrmann V. and C. Schroder., (2011), Nonlinear Eigen value and frequency response problems in industrial practice. J. Math. Ind., Vol. 1(7), pp.1–18.

37 Mehrmann V. and H. Voss., (2004), Nonlinear Eigen value problems: a challenge for modern Eigen value methods. GAMM Mitt. Ges. Angew. Math. Mech., Vol. 27(2), pp. 121–152.

38 Michiels W. and S. I. Niculescu, (2007), Stability and Stabilization of Time-Delay Systems, Volume 12 of Advances in Design and Control Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA. An Eigen value based approach.

39 Ostrowski A.M., (1958), On the convergence of the Rayleigh quotient iteration for the computation of the characteristic roots and vectors i. Arch. Rational Mech. Anal., Vol.1, pp. 233–241.

40 Ostrowski A.M., (1959), On the convergence of the Rayleigh quotient iteration for the computation of characteristic roots and vectors I pp VI, Arch. Rational. Mesh. Anal. Vol.1, pp. 233 241, Vol.2, pp. 423-428, Vol. 3, pp. 325- 340, Vol. 4, pp. 341-347, 5 pp. 472-481, Vol. 5, pp153-165.

41 Panju, M., (2011), Iterative Methods for Computing Eigen values and, Eigenvectors, University of Waterloo, arXiv preprint arXiv:1105.1185, http://mathreview.uwaterloo.ca/archive/voli/1/panju.pdf.

42 Parlett N. B., (1997), The Symmetric Eigen value Problem, Prentice-Hall, Englewood Cliffs, NJ.Reprinted as Classics in Applied Mathematics 20, SIAM, Philadelphia.

43 Saad Y., (1992), Numerical Methods for Large Eigen value Problems, Halsted Press, Div. of John Wiley & Sons, Inc., New York.

44 Sachdev S., (1999), Quantum Phase Transitions, Cambridge University Press, Cambridge.

45 Schmitt F. and Franz Rothlauf (2001), On the Mean of the Second Largest Eigen value on the Convergence Rate of Genetic Algorithms , Working Paper 1/2001, Working Papers in Information Systems Editor: Prof. Dr. Armin Heinzl.

46 Singh K.V. and Y.M. Ram, (2002), Transcendental Eigen value problem and its applications. AIAA Journal, Vol. 40(7) pp. 1402–1407.

47 Sleijpen G. L. G. and H. A. van der Vorst, (1996) A Jacobi-Davidson iteration method for linear Eigen value problems, SIAM J. Matrix Anal. Appl., 17, pp. 401–425.

48 Sorenson D. and C. Yang, (1998), A truncated RQ iteration for large scale Eigen value calculations, SIAM J. Matrix Anal. Appl., 19, pp:1045-1073.

49 Spanier J. and E. M. Gelbard Monte Carlo, (1969), Principles and Neutron Transport Problems, Addison-Wesley, Reading MA.

50 Tapia R., John Dennis, Jan Schäfermeyer, (2015) Inverse, Shifted Inverse, and Rayleigh Quotient Iteration as Newton's Method, SIAM Review In Revision, http://www.caam.rice.edu/~rat/cv/rqi.pdf.

51 Thompson J., (1972), Mathematical Statisical Mechanics, Princeton University Press, Princeton.

52 Trefethen L. N. and D. I. Bau, (1997), Numerical Linear Algebra, SIAM, Philadelphia, PA.

53 Trefethen L. N. and Mark Embree, (2005), Spectra and Pseudospectra, Princeton University Press, Princeton.

54 Unger. G., (2013), Convergence orders of iterative methods for nonlinear Eigen value problems. In Advanced Finite Element Methods and Applications, volume 66 of Lect. Notes Appl. Comput. Mech., Springer, Heidelberg, pp. 217–237.

55 Van der Sluis A. and H.A. van der Vorst, (1987) The convergence behavior of Ritz values in the presence of close Eigen values, Linear Algebra Appl., pp. 651- 694.

56 Waerden and B. L. van der, (1991), Algebra. New York ; London: Springer-Verlag. translated by Fred Blum and John R. Schulenberger. Vol.1.

57 Wigner E. P., (1950), On the statistical distribution of the widths and spacing of nuclear resonance levels. Proceedings of the Cambridge Philosophical Society,Vol. 47(4), pp. 790.

58 Wigner E. P., (1957), Statistical properties of real symmetric matrices with many dimensions. In Proceedings of 4th Canadian Mathematics Congress, pp 174-184. Princeton University.

59 Wilkinson J.H., (1965), The Algebraic Eigen value Problem, Clarendon Press, Oxford.

60 Wishart J., (1928), The generalized product moment distribution in samples from a normal multivariate population. Biometrics, 20A(1/2):pp. 32-52.

61 Xue Fei, (2009), Numerical Solution Of Eigen value Problems With Spectral Transformations, Doctor of Philosophy, University of Maryland, College Park.

62 Yadav and Vaibhav (2013), Novel Computational Methods for Solving High- mensional Random Eigen value Problems., PhD thesis, University of Iowa.

63 Yang K. (2006), Numerical algorithms for Inverse Eigen value problems arising in control and nonnegative matrices, PhD. Thesis, The Australian National University Canberra, Australia.

64 Yuan X. T. and T. Zhang, (2013), Truncated Power Method for Sparse Eigen value Problems, Journal of Machine Learning Research pp 14, Vol 899-925 (Submitted 12/11; Revised 10/12; published 4/13).