Numerical Safeguarded Use of
the Implicit Restarted Lanczos Algorithm for
Solving Nonlinear Eigenvalue Problems and
its Monotonicity Analysis

Mohammedi Radwan Hassan Abdel-Aziz

April, 1993

TR93-15
RICE UNIVERSITY
NUMERICAL SAFEGUARDED USE OF THE
IMPLICIT RESTARTED LANCZOS
ALGORITHM FOR SOLVING NONLINEAR
EIGENVALUE PROBLEMS AND ITS
MONOTONICITY ANALYSIS

by
Mohammed Radwan Hassan Abdel-Aziz

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE
Doctor of Philosophy

APPROVED, THESIS COMMITTEE:

Danny C. Sorensen, Chairman
Professor of Computational and Applied Mathematics

Richard A. Tapia
Noah Harding Professor of Computational and Applied Mathematics

Keith D. Cooper
Associate Professor of Computer Sciences, Rice University

H. Allison Smith
Assistant Professor of Engineering, Stanford University

Houston, Texas
March, 1993
Acknowledgements

This thesis would not have been possible without the encouragement and support of professor Danny Sorensen, my graduate advisor. I will always be extremely grateful for the many hours of simulating conversation and consultation he and I have had over the past few years. I am thankful to professor Richard Tapia not only for his assistance but also for his advice during all the years that I have spent at Rice University. I also thank the rest of my thesis committee professors Keith D. Cooper and H. Allison Smith for the time they have given me.

My deepest appreciation goes to my wife and to my two lovely daughters Mona and Mariam who sacrificed much in order for me to pursue my graduate studies.

I owe a great deal of thanks to my brothers, my sisters and to my reference to Rice University, Dr. Mahmoud El-Alem, Alexandria University, Egypt.

I have deeply appreciated the help of Dr. Aladin Boriek, Baylor College of Medicine, Richard Lehoucq of Rice University, Raj Singh of Stanford University, Dr. Amr El-Bakry, Alexandria University, Egypt.

Mathematical and personal thanks go to everyone in the Mathematics Department, Alexandria University, Egypt.

I would like to express my gratitude to the Center of Research and Parallel Computations, Rice University for supporting me and to Alexandria University, Egypt for giving me the opportunity to complete my graduate studies.

For keeping the department running and looking after the graduate program, I thank the computer manager Michael Pearlman, Fran Mailian and Linda Neyra.

Financial support for this research was provided in part by the National Science Foundation cooperative agreement CCR-9120008 and by the Department of Energy contract DE-FG0F-91ER25103.
NUMERICAL SAFEGUARDED USE OF THE IMPLICIT RESTARTED LANCZOS ALGORITHM FOR SOLVING NONLINEAR EIGENVALUE PROBLEMS AND ITS MONOTONICITY ANALYSIS

Mohammedi Radwan Hassan Abdel-Aziz

Abstract

In this thesis, we develop an efficient accurate numerical algorithm for evaluating a few of the smallest eigenvalues and their corresponding eigenvectors for large scale nonlinear eigenproblems. The entries of the matrices in these problems are transcendental functions approximated well by rational functions. This algorithm is based upon the Implicit Restarted Lanczos method for solving the linear eigenvalue subproblems that arise in conjunction with a new zero-finding technique that uses rational function interpolation to approximate the generalized eigenvalues. We have tested this technique on high performance computers and we present some numerical experiments that demonstrate the efficiency and the accuracy of this procedure.

Our monotonicity analysis theory shows that the parameterized eigenvalue curves (monotone increasing) are much better behaved than the parameterized determinant curves that have erratic behavior.

Our numerical and monotonicity analyses are sufficiently general that they hold for any problem having monotone increasing generalized eigenvalues. This type of problem is associated with the mixed finite element formulation that involves a frequency independent stiffness and a frequency dependent mass matrices.
TO THE MEMORY OF

MY FATHER
AND
MY MOTHER
### Contents

Abstract  \hspace{1cm} 2

List of Illustrations

1 INTRODUCTION  \hspace{1cm} 1

1.1 Some Sources of the Problem  \hspace{1cm} 1

1.2 Formulation of the Eigenvalue Problems  \hspace{1cm} 3

1.2.1 Formulation of the Mass and Stiffness Matrices  \hspace{1cm} 3

1.2.2 Finite Element Models  \hspace{1cm} 6

1.3 Mathematical Background  \hspace{1cm} 11

2 THE LINEAR EIGENPROBLEM SOLUTIONS  \hspace{1cm} 14

2.1 Lanczos Algorithm  \hspace{1cm} 14

2.2 The Computation of the Eigenpairs of the Matrix $T_k$  \hspace{1cm} 17

2.3 Relation to the $QR$ Algorithm  \hspace{1cm} 19

2.4 The Implicit Restarted Lanczos Method  \hspace{1cm} 21

3 THE NONLINEAR EIGENPROBLEM SOLUTIONS  \hspace{1cm} 24

3.1 The Formulation of the Nonlinear Eigenproblem  \hspace{1cm} 24

3.2 Historical Background  \hspace{1cm} 26

3.3 MIRL Method  \hspace{1cm} 33

4 THE MONOTONICITY ANALYSIS THEORY  \hspace{1cm} 38

4.1 Positive Definiteness for the Mixed Model  \hspace{1cm} 39

4.2 Positive Definiteness for the Exact Model  \hspace{1cm} 41
4.3 The Monotonicity Analysis

5 SOME ZERO-FINDING TECHNIQUES

5.1 Some Techniques Involving the Use of a Derivative
5.1.1 Bisection Method
5.1.2 Newton–Raphson’s Method

5.2 Some Techniques Based on Polynomial Interpolation
5.2.1 Secant Zero-finder
5.2.2 Muller’s Method

5.3 A new Technique Based on Rational Interpolation

6 THE SAFEGUARDED TECHNIQUE

6.1 The Simplified Preferable Approach

6.2 Safeguarded Rational Zero-finding Technique

6.3 The New Approach

7 NUMERICAL RESULTS

7.1 Fortran Implementation and Stopping Criterion

7.2 Comparison with Some Existing Techniques

7.3 Numerical Experiments

8 CONCLUDING REMARKS

Bibliography
Illustrations

3.1 The description of the nonlinear eigenproblem solution .................. 37
3.2 The tenth smallest eigenvalue for a nonlinear problem .................. 37

4.1 The spectrum of the mass matrix associated with the mixed finite
   element model. ........................................................................ 49
4.2 The spectrum of the mass matrix associated with the exact finite
   element model. ........................................................................ 49
4.3 The spectrum of the matrix $A(\lambda)$. .................................. 50
4.4 The spectrum of stiffness matrix associated with the exact finite
   element model. ........................................................................ 51
4.5 The spectrum of the matrix $L^{-1}(\lambda)M(\lambda)L^{-T}(\lambda)$. .......... 51

6.1 An example for a rational function ........................................... 73
6.2 The upper and lower bounds of $\mu_2$ ........................................ 73

7.1 The accumulated time for small problems ................................. 86
7.2 The accumulated time for large problems ................................. 86
Chapter 1

INTRODUCTION

This chapter consists of three sections. In the first section we present examples of algebraic eigenvalue problems arising in numerical analyses. The second section is devoted to studying the mass and stiffness matrices formulations arising from the finite element models. These formulations classify the eigenproblems into linear or nonlinear. The last section introduces from the mathematical point of view, some theorems and lemmas that are used in the monotonicity analysis proofs.

1.1 Some Sources of the Problem

The large scale eigenvalue problem often arises through the discretization of a differential operator in an attempt to approximate some of its spectral properties. However, there are many additional sources of these problems. The following are examples of these sources:

(i) Large deformation and buckling

The simplest case of buckling deformation is to consider Euler's buckling of a rod,

\[
\begin{align*}
\frac{d^2 \phi}{ds^2} + p \sin(\phi) &= 0, \\
\phi(-1) = \phi(1) &= 0
\end{align*}
\]

where, \(\phi\) is the angular deflection in points, \(b(s)\) is the stiffness in the point giving the bending moment and \(p\) is the load, with which the rod is compressed (the eigenvalue parameter \(\lambda\)). The trivial solution, \(\phi = 0\), corresponds to a straight rod. If we let \(p\) increase from zero, the rod keeps straight until \(p = \lambda_1\), where it may buckle into the first buckled state (an \(U\) shaped rod). If it does not buckle it stays straight.
until \( p = \lambda_2 \), where it may buckle into the second buckled state (an \( S \) shaped rod), and so on. In a practical situation, the occurrence of buckling is governed by small imperfections in the material or load application, leading to a more gradual onset of buckling.

(ii) The steady states of dynamical systems

As a simple case consider a chemical system

\[
c_t = f(c) + \rho \Delta c
\]

where, \( c \) is the vector of the concentration of different species, \( f(c) \) is a nonlinear function corresponding to the chemical reactions and \( \Delta c \) is the Laplace operator corresponding to the diffusion. At steady state, \( c_t = 0 \), we can study the shape of the final solution vector \( c \) for different values of the diffusivity \( \rho \) which is now used as an eigenvalue [59].

(iii) Hydrodynamic fluids

The steady axisymmetric flow of an incompressible viscous fluid between two concentric rotating circular cylinders is a well known example in this area and it is called Taylor's problem. The steady state Navier-Stokes equations govern the flow and the discretization of these equations results in a nonlinear eigenvalue problem [14].

(iv) Vibration analyses of the structural systems

This area is one of the most important sources for the eigenproblems. It is concerned with accurately predicting the free vibration modes and frequencies (the eigenvectors and the eigenvalues) of the vibrating structure. The vibration analysis process involves two general parts:

1. The finite element model of the physical system determines the model shape and frequency from the system. In addition, the model establishes the number of equations of motion (degrees-of-freedom) needed to accurately define the behavior of the vibrating system.
2. The numerical algorithm extracts the eigenvalues and their corresponding eigenvectors of the system from the resulting eigenproblem.

To maximize the effectiveness of a dynamic analysis procedure, both the finite element model and the eigensolution technique must be obtained with the most efficient use of computer resources [47]. Other examples for the sources of the eigensystems are discussed in detail elsewhere [39].

1.2 Formulation of the Eigenvalue Problems

In structural analysis we deal with discrete quantities, concentrated forces, moments, deflections and rotations at a point. Consequently, all equations of elasticity for continuous media must be reformulated as matrix equations using these discrete quantities. In these fields, two classes of matrices can be formulated, the system stiffness matrices derived from the potential energy expression and the system mass matrices derived from the kinetic energy expression.

1.2.1 Formulation of the Mass and Stiffness Matrices

In static problems, the displacements $u$ in a continuous structure can be related to a finite number of displacements $U$ selected at some arbitrary points on the structure. This relationship is expressed by the following matrix equation,

$$u = NU \quad \text{where,}$$

$$u = \{u_x, u_y, u_z\}$$

$$U = \{U_1, U_2, \ldots, U_n\}$$

$$N = N(x, y, z).$$

Formula (1.1) is valid only for small deflections. For large deflections, no such single relationship can be used in which the entries of the matrix $N$ are functions of
the coordinates only. If we use the strain-displacement, the total strain, \( \epsilon \) can be determined by

\[
\epsilon = \tilde{N} U
\]

(1.2)

where, \( \tilde{N} = \tilde{N}(x, y, z) \) is the first derivative of the matrix shape functions \( N \) for the axial case. This matrix function, \( \tilde{N} \) can be higher-order derivatives for other displacement. In dynamic problems, the relationship (1.1) is not exact, except in some special cases however, if a large enough number of displacements \( U \) are considered, the relationship \( u = NU \) will be a good approximation, provided \( U \) is determined from the dynamic equations of the system. This relationship will be applied to formulate an equivalent discrete-element system from a continuous system. To accomplish this we start with the principle of generalized virtual work that includes dynamic conditions. This generalized principle states that "in a virtual displacement of an elastic body from its instantaneous state of equilibrium, the virtual strain energy is equal to the sum of the virtual work of all the forces, including the inertia forces", [35]. This generalization may be formulated by

\[
\int_v \delta \epsilon^T \sigma dV = \int_v \delta u^T \Phi dS + \int_v \delta u^T X dV + \delta U^T P - \int_v \rho \delta \ddot{u} dV
\]

(1.3)

where,

- \( \int_v \delta \epsilon^T \sigma dV \) is the virtual strain energy, where \( \sigma \) is the total stress,
- \( \int_v \delta u^T \Phi dS \) is the virtual work of the surface forces \( \Phi = \{ \Phi_x, \Phi_y, \Phi_z \} \),
- \( \int_v \delta u^T X dV \) is the virtual work of the body forces \( X = \{ X_x, X_y, X_z \} \),
- \( \int_v \rho \delta \ddot{u} dV \) is the virtual work of the inertia forces \( \int_v \rho \ddot{u} dV \),
- \( \delta U^T P \) is the virtual work of the external forces \( P = \{ P_1, P_2, \cdots, P_n \} \) in moving through the corresponding virtual displacements \( \delta U \).

The principle of virtual work for static systems is the same expression (1.3) after eliminating the last integral. The virtual displacements \( \delta u \) and virtual strain \( \delta \epsilon \) can be obtained from equations (1.1) and (1.2) as follows.

\[
\delta u = N \delta U, \quad \delta \epsilon = \tilde{N} \delta U.
\]

(1.4)
The generalized Hook's law may be defined as

\[ \sigma = \chi \epsilon + \dot{\chi} \alpha \eta, \quad (1.5) \]

where, \( \eta \) is the temperature change, \( \alpha \) is the coefficient of the thermal expansion, \( \chi \) is a 6 \times 6 material matrix. Some of its entries depend on the Poisson ratio and the modulus of elasticity. The vector, \( \dot{\chi} \), contains entries that have the same properties as \( \chi \). We refer the reader to [35, chapter 2] for more details about this law.

After introducing equations (1.4) into expression (1.3) and using generalized Hook's law, it follows that

\[ \int_{\Omega} \delta U^T \tilde{N}^T \chi \tilde{N} U dV + \int_{\Omega} \delta U^T \tilde{N}^T \dot{\chi} \alpha dV = \int_{S} \delta u^T \Phi dS + \int_{\Omega} \delta u^T X dV + \delta U^T P - \int_{\Omega} \rho \delta \ddot{u} dV \quad (1.6) \]

Since the virtual displacements are arbitrary and \( \ddot{u} = \tilde{N} \ddot{U} \), then equation (1.6) can be written as

\[ M \ddot{U} + K U = P - \int_{\Omega} \tilde{N}^T \dot{\chi} \alpha dV + \int_{S} N^T \Phi dS + \int_{\Omega} N^T X dV \]

where,

\[ M = \int_{\Omega} \rho N^T N dV \quad \text{and} \quad K = \int_{\Omega} \tilde{N}^T \chi \tilde{N} dV \]

represent the mass and the stiffness matrices of the equivalent discrete system, respectively [35].

For an undamped linear elastic beam element with uniform properties, cross-sectional area \( \hat{A} \), mass \( \hat{m} \) per unit of its length \( \hat{l} \) and with material modulus of elasticity \( E \), these expressions take the forms

\[ m = \int_{0}^{\hat{l}} \hat{m} N^T N dx \quad \text{and} \quad k = \int_{0}^{\hat{l}} \hat{A} E \tilde{N}^T \tilde{N} dx. \quad (1.7) \]

The differential equations that rule the free vibration of that beam element are obtained by constructing the dynamic equilibrium of a small element \( \Delta x \) of the beam. In general, there are three motions occur in a two dimensional element. The axial
deformation occurs along the beam’s x-axis that is denoted by \( u \), the lateral vibration in the y-direction and is indicated by \( v \) and the rotation \( \theta \) around z-axis. The equilibrium equations are found in the horizontal direction for axial vibration and in the vertical direction for lateral vibration. It is assumed that the axial and lateral vibration behaviors are independent. Consequently, there is no coupling between \( u \) and \( v \). Using Hook’s law and elementary beam theory, the equations governing axial and lateral vibrations take the following forms:

\[
\frac{\partial^2 u}{\partial x^2} - \frac{m}{AE} \frac{\partial^2 u}{\partial t^2} = 0, \tag{1.8}
\]

and

\[
\frac{\partial^4 v}{\partial x^4} + \frac{m}{EI} \frac{\partial^2 v}{\partial t^2} = 0, \tag{1.9}
\]

respectively. Where, \( I \) is the beam moment of inertia about z-axis. It is quite easy to show that the partial differential equations of the types (1.8) and (1.9) possess nontrivial solutions only if the quantities \( \frac{m}{AE} \) and \( \frac{m}{EI} \) are positive, which indeed is the case for structural dynamics. The solution for this equation can be formulated by separating the variables and imposing the displacement boundary conditions and the force boundary conditions. An approximated solution for these vibrations can be obtained by using the finite element methods [21].

1.2.2 Finite Element Models

The dynamic response of simple structures, such as uniform rod and beam elements, may be obtained by solving the differential equations of motion together with appropriate boundary conditions. In many practical situations, either the geometrical or material properties vary. In addition, the boundaries may not be described in terms of known functions. Moreover, practical structures consist of an assemblage of components of different types, namely, plates, shells and beams. In these situations it is generally impossible to obtain analytical solutions to the equations of motion which satisfy the boundary conditions. This difficulty is overcome by seeking approximate
solutions. There are a number of techniques available for determining these approximate solutions. One of the most widely used procedures is the finite displacement element method, in which, the element displacement field represents the solution to the differential equation of motion that governs the system. The shape function matrices are derived from element displacement fields and can be expressed through series expression in ascending powers of $\omega$. This expression takes the form

$$N(\omega) = N_0 + \omega N_1 + \omega^2 N_2 + \omega^3 N_3 + \cdots$$

where, $N(\omega)$ is the exact frequency dependent shape function matrix based on the solution to the differential equation of motion, $N_0$ is the shape function matrix based on the static displacement field and $N_1, N_2, N_3, \cdots$ are the shape function matrices representing the dynamic corrections [35]. There are three classes of finite element models which are capable of discretizing the field, the conventional model, the exact displacement model and the formulated (mixed) exact model [23]. The shape function matrix is based on the assumed element displacement field. Choice of this field plays a crucial role in both the accuracy and efficiency of the solution.

(a) The conventional Finite Element Model

Separation of variables, $u(x,t) = \hat{u}(x)g(t)$, can be used to reduce the equation of motion for axial vibration, equation (1.8), in two ordinary differential equations. One of the two depends on the position, $x$ and the other one depends on the time $t$; specifically, the spatial vibration is governed by

$$\frac{d^2 \hat{u}}{dx^2} + \omega^2 \frac{\bar{m}}{AE} \hat{u}(x) = 0. \quad (1.10)$$

The linear finite element model uses polynomial shape functions to approximate the solution of equation (1.10), namely, $\hat{u}(x) \approx \hat{u}_p(x) = N_p q$ where, the subscript $p$ denotes that the approximation was formulated using the polynomial based shape function $N_p$ and $q$ is the vector of nodal degrees-of-freedom based on the polynomial element displacement. The shape function matrix $N_p = [(1 - \xi) \quad \xi]$, where $\xi = \xi_i$, with
expressions (1.7) are used to define the elemental stiffness matrix, \( k_p \) and elemental mass matrix, \( m_p \) as follows.

\[
k_p = \frac{\hat{A}E}{l} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \quad \text{and} \quad m_p = \frac{\hat{m}l}{6} \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}
\]  

(1.11)

**Remark**: The sum of the entries in each row of the elemental stiffness matrix is zero. This indicates that when the element moves as a rigid body, the elastic restoring forces are zero and the sum of the entries in the elemental mass matrix is \( \hat{m}l \), which is the total mass of the element.

The assembly process for a beam element is similar to that of an axial element. For element \( i \) with nodes \( i \) and \( (i + 1) \), the two rows and columns of the elemental mass matrix \( m_p \) are added into rows and columns \( i \) to \( (i + 1) \) of the mass matrix for the complete beam. The mass matrix resulting from the assembly process of \( m_p \) is called the global mass matrix \( M \), [33]. The elemental stiffness matrix \( k_p \) is treated in the same way and we end up with the global stiffness matrix \( K \). The global stiffness and mass matrices are symmetric positive definite after deleting the rows and their corresponding columns that associate with the boundary conditions. These global matrices define the linear eigenvalue problem

\[
[K - \omega^2 M]v = 0
\]

where, \( v \) is the associated eigenvector of \( n \) components representing the model displacements. Because \( N_p \) is based on the displacement field for static analyses, the associated systems matrices can be exact for static analyses, but approximate for dynamic analyses.

Linear finite element models involve discretization error due to the use of approximate displacement fields in representing element deformations. Therefore, the eigensolutions obtained from linear finite element eigenproblems are exact only as the number of degrees-of-freedom (representing the matrix order) approaches infinity and higher-order modes become increasingly inaccurate for a given model, [4]. Vibration
analyses based on the linear finite element model becomes computationally intensive due to the large number of degrees-of-freedom needed for the high accuracy solution [47].

(b) The Exact Finite Element Model

The static displacements in a uniform beam (undergoing axial vibration) have been prescribed by displacements based on the polynomial displacement field. In the exact model, we are using exact shape functions. Suppose that a uniform bar of length \( l \) performs harmonic forced vibrations by varying the end displacements so that the equation of motion,

\[
\frac{c^2}{2} \frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial t^2} = 0 \tag{1.12}
\]

with boundary conditions,

\[
u(0, t) = \beta_1 e^{i\omega t}, \quad u(l, t) = \beta_2 e^{i\omega t} \tag{1.13}
\]

governs the motion. The constants \( \beta_1 \) and \( \beta_2 \) represent the displacement amplitudes and \( c^2 = \frac{4E}{m} \). The displacement in the bar is assumed as follows:

\[
u = h(x) e^{i\omega t} \tag{1.14}
\]

where, \( h(x) \) is a function of \( x \) only. Introducing formula (1.14) into equation (1.12) and using the boundary conditions (1.13) we have

\[
u(x) = \begin{pmatrix}
\cos \nu x - \sin \nu x \frac{\cos \nu \beta_1}{\sin \nu} \\
\sin \nu x \frac{\sin \nu \beta_1}{\sin \nu}
\end{pmatrix}
\begin{pmatrix}
\beta_1 \\
\beta_2
\end{pmatrix}
\begin{pmatrix}
e^{i\omega t}
\end{pmatrix} \tag{1.15}
\]

where \( \nu = \sqrt{\frac{E}{\mu}} \omega \) and \( \nu \neq n\pi \). When \( \nu = n\pi \), the forcing frequency is equal to the natural frequency of a free bar of length \( l \), and the displacement \( u \) becomes infinite.

Formulas (1.13) indicate that for harmonic boundary conditions the displacement \( u \) depends on the instantaneous values of the boundary displacement \( u(0, t) \) and \( u(l, t) \). Thus, in this case the displacements at any point along the length of the bar can be
related to the boundary displacement through the relationship, \( u = N_e(\nu)q \), where

\[
N_e(\nu) = \begin{pmatrix}
\cos \nu x - \sin \nu x \cos \nu l \\
\sin \nu x \sin \nu l
\end{pmatrix}
\begin{pmatrix}
\sin \nu x \\
\sin \nu l
\end{pmatrix}, \quad q = \begin{pmatrix}
\beta_1 \\
\beta_2
\end{pmatrix} e^{i\omega t}
\]

and the subscript \( e \) denotes that we have used exact displacements. This relationship is true for any structure or structural element performing forced or free harmonic motion. After substituting with \( N_e(\nu) \) into expression (1.7) and integrating over the volume of the bar element, it can be shown that the elemental mass matrix is given by

\[
m_e(\nu) = \frac{m l}{2} \begin{pmatrix}
\frac{1}{\sin^2 \nu l} & \frac{\nu l \sin \nu l}{\sin^2 \nu l} & \frac{\cos \nu l}{\sin \nu l} \\
\frac{\nu l \sin \nu l}{\sin^2 \nu l} & \frac{1}{\sin^2 \nu l} & -\frac{\cos \nu l}{\sin \nu l}
\end{pmatrix}
\]

(1.16)

and the elemental stiffness matrix \( k_e(\nu) \) can be found by using expressions (1.7) and the first derivative of the function \( N_e(\nu) \) as follows:

\[
k_e(\nu) = \frac{A E}{2l} \begin{pmatrix}
\frac{(\nu l)^2}{\sin^2 \nu l} & \frac{\nu l \cos \nu l}{\sin \nu l} \\
\frac{\nu l \cos \nu l}{\sin \nu l} & -\left( \frac{\nu l}{\sin \nu l} + \frac{(\nu l)^2 \cos \nu l}{\sin^2 \nu l} \right)
\end{pmatrix}
\]

(1.17)

The assembly process of a rod element is the same as an axial element. For element \( i \) with nodes \( i \) and \( (i + 1) \), the two rows and columns of the elemental mass matrix (1.16) are added into rows and columns \( i \) to \( (i + 1) \) of the mass matrix for a complete bar [33]. The stiffness matrix is treated in the same way. This assembly process leads us to define the nonlinear eigenvalue problem:

\[
[K(\omega) - \omega^2 M(\omega)]v = 0.
\]

(1.18)

Although fewer degrees-of-freedom are associated with the exact displacement model than with the linear finite element model, the reformulation and redecomposition of the matrices for each iteration of the eigenvalue analysis cause the exact model to be computationally intensive [23].

(c) The Mixed Finite Element Formulation

The new formulation discussed by Melosh and Smith in [23] uses both the exact
and polynomial displacement fields, $N_e(\nu)$ and $N_p$ respectively, in formulating the vibration problem. This formulation is based on exact representation of element inertia forces and obtains exact natural frequencies and their corresponding modes of the vibrating systems. The elemental mass and stiffness matrices are defined by the expressions

$$m_x(\nu) = \int_0^1 \dot{m} N_p^T N_e(\nu) dx, \quad k_p = \int_0^1 \dot{A} E N_p^T N_p dx.$$  

(1.19)

Using expression (1.19) and the previous definitions of $N_p$ and $N_e(\nu)$, the elemental stiffness matrix $k_p$ takes the form as in formulas (1.11) and the elemental mass matrix can be defined as follows:

$$m_x(\nu) = \hat{m} l \begin{pmatrix} \frac{1}{(\nu l)^2} - \frac{\cos \nu l}{\nu l \sin \nu l} & \frac{1}{\nu l \sin \nu l} - \frac{1}{(\nu l)^2} \\ \frac{1}{\nu l \sin \nu l} - \frac{1}{(\nu l)^2} & \frac{1}{(\nu l)^2} - \frac{\cos \nu l}{\nu l \sin \nu l} \end{pmatrix}.$$  

(1.20)

The subscript $x$ denotes that we have used mixed finite element for generating the elemental mass matrix. The assembly process for element $i$ with nodes $i$ and $(i+1)$ (for axial vibrations only) may be described as follows:

The two rows and columns of the mass matrix (1.20) are added into rows and columns $i$ to $(i+1)$ of the mass matrix for a complete beam. This leads to the global mass matrix $M(\nu)$. By using the elemental stiffness matrix we will end up with the global stiffness matrix $K$. These global mass and stiffness matrices are sparse and symmetric and they define the following nonlinear eigenvalue problem

$$[K - \omega^2 M(\omega)] v = 0.$$  

(1.21)

When a high accuracy solution is required, the mixed finite element model is more computationally efficient than the linear finite element and the exact displacement models.

1.3 Mathematical Background

The symmetric matrices have a theory that is both rich and elegant. Some of the important aspects of this theory are summarized in this section. A class of symmetric
matrices with a special positivity property arises naturally in many applications. Symmetric matrices with this positivity property also provide one generalization to matrices of the notion of a positive number. This observation often provides insight into the properties and applications of positive definite matrices. If the symmetric form \((Mv,v) = v^T M v\) is positive (nonnegative) for all non-zero \(v \in \mathbb{R}^n\), then we call this positive definite (positive semi-definite) symmetric form, and we call the symmetric matrix \(M = [a_{ij}]\) a positive definite (positive semi-definite) matrix. The matrix \(M\) is said to be diagonally dominant if

\[|a_{ij}| \geq \sum_{j=1}^{n} |a_{ij}|, \text{ for all } \ i \neq j \text{ and } i = 1, 2, \cdots n.\]

If this inequality is strict inequality then this matrix is said to be strictly diagonally dominant. The following lemmas and theorems give some qualities of the symmetric positive definite matrices and the proofs of them can be found in [13, 9, 49].

**Lemma 1.1** All the eigenvalues of a symmetric matrix are positive if and only if this symmetric matrix is positive definite.

**Theorem 1.1** A symmetric matrix is positive definite if and only if all its principal minors are positive.

**Theorem 1.2** Every symmetric positive definite matrix \(K \in \mathbb{R}^{n \times n}\) can be factorized using Cholesky decomposition into \(LL^T\) where, \(L\) be a unit lower triangular matrix.

**Lemma 1.2** If \(M\) is a symmetric positive definite matrix and \(L\) is a unit lower triangular matrix then \(L^{-1}ML^{-T}\) is a symmetric positive definite matrix.

**Theorem 1.3** (Gersgorin theorem) If \(M \in \mathbb{R}^{n \times n}\) is a symmetric matrix then each eigenvalue of \(M\) lies in one of the disks in the real plane.

\[D_i = \{ \lambda : |\lambda - a_{ij}| \leq \sum_{j=1}^{n} |a_{ij}|, i \neq j \} \quad (i = 1, 2, \cdots n)\]
Theorem 1.4 (Courant–Fischer Minimax Theorem)

Let the symmetric matrix \( M \in \mathbb{R}^{n \times n} \) have eigenvalues \( \lambda_1, \lambda_2, \ldots, \lambda_n \) ordered so that \( \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n \), then

\[
\lambda_k = \min_{\dim(X) = k} \max_{0 \neq v \in \mathbb{R}^n} \frac{v^T M v}{v^T v} = \max_{\dim(X) = n-k+1} \min_{0 \neq v \in \mathbb{R}^n} \frac{v^T M v}{v^T v}
\]

where, \( k = 1, 2 \cdots n \) and \( \lambda_k(M) \) denote the \( k \)-th largest eigenvalue of \( M \).

Lemma 1.3 Let \( M \) be a symmetric strictly diagonally dominant matrix with positive diagonal entries then \( M \) is positive definite.

Definition 1.1: A function \( \mu : S \subset \mathbb{R}^n \to \mathbb{R} \) is convex on a convex set \( S_0 \subset S \) if for all \( x_1, x_2 \in S_0 \) and \( 0 < \theta < 1 \)

\[
\mu(\theta x_1 + (1 - \theta)x_2) \leq \theta \mu(x_1) + (1 - \theta) \mu(x_2).
\]

(1.22)

If the inequality (1.22) is strict inequality, then \( \mu \) is strictly convex on \( S_0 \). If strict inequality in (1.22) holds whenever \( x_1 \neq x_2 \), then \( \mu \) is uniformly convex on \( S_0 \) if there is a constant \( \beta > 0 \) such that, for all \( x_1, x_2 \in S_0 \) and \( 0 < \theta < 1 \)

\[
\theta \mu(x_1) + (1 - \theta) \mu(x_2) - \mu(\theta x_1 + (1 - \theta)x_2) \geq \beta \theta (1 - \theta) \| x_1 - x_2 \|.
\]

For more details about convexity and the relationship between convexity and differentiation see [28, 52].
Chapter 2

THE LINEAR EIGENPROBLEM SOLUTIONS

The approach that is taken in this thesis for solving nonlinear eigenproblems is based on solving a sequence of linear eigenvalue subproblems. The linear eigenproblems may be nonsymmetric or symmetric. The Arnoldi process [1] is an appropriate technique for finding a few of eigenvalues and their corresponding eigenvectors of a large nonsymmetric square matrix. For large symmetric eigenproblems, we use a variant of Lanczos algorithm. There are also many methods that depend on similarity transformations to solve the linear eigenvalue problems, but these methods are not useful for large problems [9, 49, 54].

This chapter consists of three parts. In the first part, we propose the Lanczos algorithm and some of its advantages. The second part introduces the approximation of the eigenpairs of the original problem and the relationship between the Lanczos method and the $QR$ algorithm. The last part is devoted to studying in detail a complete iteration of the Implicit Restarted Lanczos method (IRL) with exact shift strategy.

2.1 Lanczos Algorithm

The Lanczos algorithm for computing eigenvalues and eigenvectors for a large scale symmetric eigenproblem $Ax = \lambda x$ was introduced in the early 1950’s by Lanczos [20]. It begins with the specification of the starting vector $v_1 \in R^n$, and an $n \times n$ real symmetric matrix $A$. After taking $k$-steps, the Lanczos process produces an orthonormal $n \times k$ matrix $V$, a symmetric tridiagonal matrix $T$ and an $n$-vector $f$
with $V^T f = 0$, such that

$$AV = VT + f e_k^T$$  \hspace{1cm} (2.1)

where, the vector $f$ is called the residual vector and $e_k$ is the $k$-th coordinate basis vector. The generalized eigenvalue problem $Ax = \lambda Bx$, where $B$ is symmetric positive definite, may be addressed by constructing the factorization

$$AV = BVT + f e_k^T$$

where, $V^T B V = I$ is maintained. The approximated eigenvalues and eigenvectors are readily available through this factorization. In particular, this approximation is exact when $f = 0$.

The principle of the Lanczos process is that it is a projection technique on a Krylov subspace. The major practical differences with the non-hermitian methods are that the matrix $A$ is reduced to a tridiagonal matrix, $T$ and, more importantly, that we only need to save three vectors, at least if we do not restore to any form of reorthogonalization. In addition, the largest and smallest eigenvalues will emerge well before the tridiagonalization is completed. Because of these features, the Lanczos method is especially attractive for applications when a few of the largest or the smallest eigenvalues are of interest. Unfortunately, the Lanczos process has some numerical difficulties resulting from the loss of orthogonality among the Lanczos vectors, the appearance of spurious eigenvalues in the spectrum of $T$, the error due to finite precision arithmetic, and the undetermined storage requirement. These drawbacks have been the subject of considerable research over the last two decades [9, 31, 38, 6]. For sake of clarity in this study, the Lanczos iteration approach will be defined in the next algorithm. Suppose the Lanczos process is applied for $k$-steps then it can continue further $p$-steps by using the following algorithm

**Algorithm 2.1**

function $[T, V, f] = \text{Lanczos}(A, T, V, f, k, p)$

Input: $AV - VT = f e_k^T$ with $V^T V = I_k$, $V^T f = 0$. 
Output: $AV - VT = f e_{k+p}^T$ with $V^TV = I_{k+p}$, $V^T f = 0$.

(1) For $j = 1, 2, \ldots, p$

1. $\beta \leftarrow \| f \|_2$; if $\beta < tol$ then stop;
2. $T \leftarrow \begin{pmatrix} T & \beta e_{k+j-1}^T \\ \beta e_{k+j-1} & \alpha \end{pmatrix}$; $v \leftarrow \frac{1}{\beta} f$; $V \leftarrow [V, v]$;
3. $w \leftarrow Av$;
4. $\alpha \leftarrow v^T w$; $T \leftarrow \begin{pmatrix} T & \beta e_j^T \\ \beta e_j & \alpha \end{pmatrix}$;
5. $f \leftarrow w - \alpha v - \beta v_-$;
6. while $\| s \| > \epsilon \| f \|_2$;
   a. $s = V^T f$;
   b. $f \leftarrow f - Vs$;
   c. $\alpha \leftarrow \alpha + \sigma_j$;
   $$\beta \leftarrow \beta + \sigma_{j-1}$$

Remark 1: The vector $v_-$ is the column before the last one of the matrix $(V, v)$, i.e. $v_- = (V, v)e_{k+j-2}$ and $\sigma_j$ is the j-th component of the vector $s$. Steps (1.6) assure orthogonality with iterative refinement, for more details of the implementation see [31, 16]. Computational experience with this device indicates that it is sufficient to do at most two steps of iterative refinement, [45].

Remark 2: The sequence in the steps 4 and 5 can be described in terms of polynomials which are easily obtained by considering the principal minors of the tridiagonal matrix $[\lambda I - T]$ as.

$$P_0(\lambda) = 1$$
$$\beta_2 P_1(\lambda) = (\lambda - \alpha_1)$$
$$\beta_3 P_2(\lambda) = (\lambda - \alpha_2) P_1(\lambda) - \beta_2 P_0(\lambda)$$
$$\vdots$$
$$\beta_{i+1} P_i(\lambda) = (\lambda - \alpha_i) P_{i-1}(\lambda) - \beta_i P_{i-2}(\lambda)$$
Lanczos stated, [20]: The correct eigenvalues of the matrix $A$ are obtained by evaluating the zeros of the last polynomial $P_n(\lambda) = 0$. What actually happens, however, is that the zeros of the polynomials $P_i(\lambda)$ do not change much from the beginning. If the dispersion (of the eigenvalues) is strong, then each new polynomial basically adds one more root, but corrects the higher roots by only small amounts. It is thus well possible that the series of largest roots, in which we are primarily interested, is practically established with sufficient accuracy after a few iterations. Then we can stop, since the latter iterations will change the obtained values by negligible amounts. The same can be said about the vibrational modes associated with these roots. Thus the Lanczos algorithm can be viewed as a method for the construction of the characteristic polynomials in progressive steps, and the eigenvalues of the $k \times k$ matrix $T_k$ are merely the roots of the polynomial $P_k(\lambda) = 0$.

2.2 The Computation of the Eigenpairs of the Matrix $T_k$

The underlying principles of the foregoing truncation process become clearer when the process is shown to be nothing more than application of the Rayleigh-Ritz technique. In Lanczos method, $k$ assumed displacement functions of the form $X_k = [x_1, x_2, \cdots, x_k]$ are taken to be linear combinations of the Lanczos vectors $V_k$, so that

$$X_k = V_k Y_k$$  \hspace{1cm} (2.2)

where, $Y_k = [y_1, y_2, \cdots, y_k]$ are the Ritz-coordinates which are to be determined. Forming the $k \times k$ Rayleigh quotient gives

$$\rho(X_k) = \frac{Y_k^T V_k^T A V_k Y_k}{Y_k^T V_k Y_k} = \frac{Y_k^T T_k Y_k}{Y_k^T Y_k}$$

whose $k$ stationary values are given by the eigensolution

$$T_k Y_k = Y_k \Theta \quad Y_k^T Y_k = I$$

where, $\rho(X_k) = \Theta = \text{diag}(\theta_1, \theta_2, \cdots, \theta_k)$ and $\theta_1 \geq \theta_2 \geq \cdots \geq \theta_k$. The eigenvectors $Y_k$ define the Ritz coordinates or coefficients, when combined with the $V_k$. The Ritz
vectors $X_k$ in equation (2.2) and the corresponding Ritz values $\Theta$ are the optimum estimates obtainable from the space spanned by the Lanczos vectors $V_k$. The solution of the truncated tridiagonal eigenvalue problem

$$T_kY_k = Y_k \Theta_k$$

(2.3)

can be obtained efficiently by a variety of methods because $T_k$ is symmetric tridiagonal and will normally be of small order. By postmultiplying equation (2.1) by $Y_k$, the $k \times k$ matrix of Ritz eigenvectors gives

$$AV_kY_k = V_kT_kY_k + f e_k^T Y_k.$$  

(2.4)

Substituting from equations (2.2) and (2.3) into equation (2.4) gives

$$AX_k = X_k \Theta_k + f e_k^T Y_k.$$  

Examining any vector $x_j$ leads to the form

$$Ax_j = x_j \theta_j + f y_{kj}$$

where, $y_{kj} = e_k^T Y_k e_j$ is the $(k,j)$ element of the matrix $Y_k$. Taking norms of the both sides then yields

$$|\lambda_j - \theta_j| = \|f\| |y_{kj}|$$

(2.5)

since $\|x_j\| = 1$ and $\|f\| = \beta$. Thus knowing $\beta$ and the bottom row of the eigenvector matrix $Y_k$, the error bounds relating the closeness of the $k$ approximates Ritz values $\theta_j$ ($j = 1, 2, \ldots, k$) to the actual eigenvalues $\lambda_j$ ($j = 1, 2, \ldots, k$) can be obtained. The error bounds are equivalent to the residual norm $\|Ax_j - \lambda_j x_j\|_2$ and therefore it is a very economical estimation, since the norm can be calculated without the explicit formulation of $x_j = V_k y_j$ or $A$, both of which involve expensive calculations in terms of space and operations. Parlett [31] employs equation (2.5) to signal convergence by assuming the $k$ Ritz values to have converged when

$$|\lambda_j - \theta_j| \leq \frac{\|f_j\|^2}{C(\theta_j)} \leq \epsilon \max_i \{|\theta_j|\}, \quad j = 1, 2, \ldots, k$$
where, $C(\eta) = \min\{[(\theta - \eta) : \theta \in \sigma(T)]\}$ is the gap condition, $\|f_j\|$ is the residual norm, $\sigma(T)$ is the spectrum of $T$, the choice of $\max_i$ is to make it scale independent and $\epsilon$ is a "suitably small" value defined as being equal to $\sqrt{\epsilon_1}A\|_2$, where $\epsilon_1 = \text{macheps}$. The quantity macheps is the machine’s floating-point precision (a very small quantity attainable on the computer such that $\text{macheps} + 1 > 1$). The norm $\|A\|_2 = \|T\|_2 = |\theta_1| \approx |\lambda_1|$, so that $\epsilon = \sqrt{\epsilon_1}|\lambda_1|$ can be calculated readily after the first few steps of a Lanczos run. When $\theta_1$ will have converged to $\lambda_1$, the choice of $\epsilon$ is based on the recommendations of Parlett.

The outcome of this discussion is when $(\theta, y)$ is the eigenpair for $T$, (i.e $Ty = \theta y$), then the pair $(\theta, x)$ is an approximated eigenpair for $A$ where, $x = Vy$. In other words let $AV_k - V_kT_k = f_k e_k^T$ be a $k$-step Lanczos factorization of $A$, with $f_j \neq 0$, $0 \leq j \leq k - 1$. Then $f_k = 0$ and $T_k$ is diagonalizable if and only if $v_1 = \sum_{i=1}^{k} x_i$, where $\{x_i\}$ is a set of $k$ linearly independent eigenvectors for $A$.

### 2.3 Relation to the QR Algorithm

In order to motivate the ideas in the remainder of this chapter, it will be instructive to derive and analyze the QR iteration from a certain point of view [46]. To do this, suppose that there has been a complete reduction of $A$ to tridiagonal form. Thus

$$AV - VT = 0, \quad \text{where} \quad V^TV = I_n \quad \text{and} \quad T - \text{tridiagonal}. \tag{2.6}$$

Let $\gamma$ be the shift and let $[T - \gamma I] = QR$ with $Q$ orthogonal and $R$ upper triangular. The explicitly shifted $QR$ algorithm consists of the following four steps

1. $[A - \gamma I]V = V[T - \gamma I] = 0$
2. $[A - \gamma I]V = V[QR] = 0$
3. $[A - \gamma I][VQ] - [VQ][RQ] = 0$
4. $A[VQ] - [VQ][RQ + \gamma I] = 0.$
After these four steps we have updated formula (2.6) to produce

\[ AV_+ - V_+ T_+ = 0 \]  \hspace{1cm} (2.7)

where, \( V_+ = VQ \), and \( T_+ = RQ + \gamma I \) is tridiagonal. Note that from step (3), it follows that

\[ [A - \gamma I]v_1^+ = v_1^+ \rho_{11} \quad \text{where,} \quad \rho_{11} = e_1^TRe_1, \quad v_1^+ = V_+ e_1. \]

Moreover, formula (3) gives

\[ [VQ]^{-1} [A - \gamma I]^{-1} - [RQ]^{-1} [VQ]^{-1} = 0. \]

Hence

\[ Q^T V [A - \gamma I]^{-1} - Q^T R^{-1} V_+^T = 0, \]

i.e

\[ V^T [A - \gamma I]^{-1} - R^{-1} V_+^T = 0 \]

so that

\[ [A - \gamma I]^{-1} v_n = \rho_{nn} v_n^+ \]

where, \( v_n = V e_n \), \( v_n^+ = V_+ e_n \) and \( \rho_{nn} = e_n^T R^{-1} e_n \). This proves that the QR iteration is performing inverse iteration [49] with respect to \( A^T \) on the last column of \( V \).

An implicitly shifted QR step starting with equation (2.6) consists of

\[ A[VQ] - [VQ][Q^T Q] = 0 \]

where the orthogonal matrix \( Q \) is computed as a product of Givens transformation which are specified implicitly through the well known “bulge chase” sequence as described in [31, 49] once the shift \( \gamma \) is specified. From the previous discussion, the application of \( p \) implicit shifts will result in the implicit application of a polynomial \( \psi \) of degree \( p \) to the vector \( v_1 \). Thus once the \( p \) shifts have been applied

\[ AV_+ - V_+ T_+ = 0 \]
where, \( V_+ = V_1Q_2 \cdots Q_p \), \( T_+ = Q_1^T \cdots Q_p^T Q_1 T Q_2 \cdots Q_p \) with \( v_1^+ \equiv V_+ e_1 \) satisfying

\[ v_1^+ = \psi(A)v_1 \]

where, \( \psi(\lambda) = \frac{1}{\nu} \prod_{j=1}^{p} (\lambda - \gamma_j) \) with \( \nu \) a normalizing factor to make \( \|v_1^+\| = 1 \) and \( \{\gamma_j\} \) the set of \( p \) implicit shifts.

From this point of view, one may interpret the QR iteration as a process of rapidly determining an approximate root \( \gamma \) of the characteristic polynomial and then applying the linear factor \( A - \gamma I \) to \( v_1 \) to replace it with \( v_1^+ \leftarrow \frac{1}{\nu} (A - \gamma I)v_1 \) in order to purge the starting vector components along eigenvectors associated with \( \gamma \). As the iteration proceeds, subdiagonal elements of \( T \) must tend to zero.

### 2.4 The Implicit Restarted Lanczos Method

The computational difficulties mentioned above stem from the fact that the residual \( \|f\| = 0 \) if and only if the columns of \( V \) span an invariant subspace of \( A \). When \( V \) "nearly" spans such a subspace, \( \|f\| \) will be small. Typically, in this situation, a loss of significant digits will take place. These issues can be controlled by limiting the number of steps in the Lanczos process to a certain value \( k \) and then treating the residual vector as a function of the initial Lanczos vector. This starting vector is then updated through an iterative scheme (polynomial filters) which is designed to force convergence of the residual to zero. This approach was developed by Sorensen in [46]. With the standard Lanczos defined in algorithm 2.1 and its relationship with the QR algorithm, it is possible to describe the complete iteration of Implicit Restarted Lanczos method in the following algorithm.

**Algorithm 2.2**

function \([T, V, f] = IRL(A, k, p, tol)\).

1. initialize \( V(:, 1) = v_1; T \leftarrow (v_1^T A v_1); f \leftarrow A v_1 - v_1 T \);
2. \([T, V, f] \leftarrow \text{Lanczos}(A, T, V, f, k, 1)\)
(3) For \( m = 1, 2, \ldots \) until convergence

(a) if \( \|f\| < tol \) then stop;

(b) \([T, V, f] \leftarrow \text{Lanczos}(A, T, V, f, k, p)\);

(c) \( h = \text{Shifts}(T, p) \);

(d) \( Q \leftarrow I_{k+p} \);

(e) for \( j = 1, 2, \ldots, p \)

\( (e.1) \quad T \leftarrow Q_j^T T Q_j \); (Bulge-Chase with shift \( \gamma_j = h(j) \))

\( (e.2) \quad Q \leftarrow QQ_j \);

(4) \( v \leftarrow [VQ] e_{k+1} \); \( V \leftarrow [VQ] \begin{pmatrix} I_k \\ 0 \end{pmatrix} \);

(5) \( f \leftarrow (v \beta_k + f \sigma_k) \); where \( \beta_k = e_{k+1}^T T e_k \), \( \sigma_k = e_{k+p}^T Q e_k \);

Remarks:

(i) The Bulge Chase at step (3.e.1) is defined implicitly as usual so that \( T - \gamma_j I = Q_j^T R_j \); since all the eigenvalues are real there is no need for applying the implicit double shift strategy.

(ii) At step (3.e.1), it may happen that a sub-diagonal element \( \beta_j \) becomes small.

The deflation strategies associated with the QR algorithm are then employed.

In case (ii), the matrix \( T \) is split giving

\[
T = \begin{pmatrix} T_j & \beta e_j e_1^T \\ \beta e_1 e_j^T & \hat{T}_j \end{pmatrix} \approx \begin{pmatrix} T_j & 0 \\ 0 & \hat{T}_j \end{pmatrix}, \quad VQ = [V_j, \hat{V}_j].
\]

Thus, an invariant subspace of dimension \( j \) has been found. If \( j \geq k \) and all the shifts have been applied then the iteration is halted. Otherwise \( T_j, V_j \) are retained and the iteration proceeds with \( \hat{V}_j, \hat{T}_j \) filling the role of \( V, T \) respectively. However, \( T_j \) continues to participate in the shift selection strategy on subsequent iterations. That is, all of the eigenvalues of \( T \) are considered in the selection process. If some
of the eigenvalues of $T_j$ are selected as shifts then these are applied implicitly to $T_j$ to split this matrix and the unwanted portion is discarded to form a submatrix of smaller size. The corresponding columns of the (updated) matrix $V_j$ are discarded and then $\tilde{V}_j$ and $\tilde{T}_j$ are shifted to left. The remaining shifts are applied implicitly to $\tilde{T}_j$ and then the Lanczos factorization is completed to fill out the remainder of the $k + p$ columns of $V$. In this way the iteration is not terminated by deflation until the appropriate approximation to the wanted spectrum has appeared. The application of the implicit shift $\gamma_j$ will replace the starting vector $v_1$ with $[A - \gamma_j I]v_1$. Thus after completion of each cycle of the loop at step 3 in algorithm (2.2):

$$V e_1 = v_1 \leftarrow \psi(A)v_1,$$

where $\psi(\lambda) = \frac{1}{p} \prod_{j=1}^{p} (\lambda - \gamma_j)$. Numerous choices are possible for the selection of these $p$ shifts. One possibility is choosing $p$ "exact" shifts with respect to $T$. Thus the selection process might be describe as in the following algorithm

**Algorithm 2.3**

function $[h] = \text{Shifts}(T, p)$

1. **Compute** $\lambda(T)$ (by QR for example)

2. **Select** $p$ unwanted eigenvalues $\{h(j) \leftarrow \gamma_j : 1 \leq j \leq p\} \subset \lambda(T)$

Some obvious criterion for this selection might be

1. Sort $\lambda(T)$ according to algebraically largest eigenvalues and select the $p$ smallest eigenvalues as shifts;

2. Sort $\lambda(T)$ according to largest modulus and choose the smallest $p$ eigenvalues as shifts;

Selecting these exact shifts has interesting consequences in the iteration. For details and alternative strategies for choosing shifts see [46].
Chapter 3
THE NONLINEAR EIGENPROBLEM SOLUTIONS

The techniques described in chapter 2 are effective in the solution of the linear eigenproblems. They may be used to produce effective methods for solving the parametric and the nonlinear eigenvalue problems through iterative schemes.

This chapter consists of three parts. In the first part, we define the general nonlinear eigenvalue problem and some of its special cases. The second part is devoted to presenting from a historical point of view some techniques that attempt to solve the nonlinear eigenproblem. The last part introduces the MIRL method which we shall use in our algorithm.

3.1 The Formulation of the Nonlinear Eigenproblem

The most general formulation of the nonlinear eigenvalue problem is to find $\lambda, x$ and $y$ such that:

\[
[G(\lambda)]x = 0, \\
y^h[G(\lambda)] = 0
\]

where, $[G(\lambda)]$ is a matrix valued function of the eigenvalue $\lambda$, $x$ and $y$ are the right and the left eigenvectors corresponding to that eigenvalue, respectively. The linear eigenproblem is a special case from this general formula (3.1) if we define

\[
G(\lambda) = \lambda I - M \quad \text{or} \quad G(\lambda) = \lambda K - M.
\]

The quadratic case:

\[
G(\lambda) = \lambda^2 A_2 + \lambda A_1 + A_0.
\]
arises from the solution of Lagrange’s equations of motion for non-conservative mechanical systems. The reader is referred to [19, 15] for a discussion of that subject.

In the case where \( G(\lambda) \) is a polynomial,

\[
G(\lambda) = \lambda^j A_j + \lambda^{j-1} A_{j-1} + \ldots + A_0. \tag{3.2}
\]

The standard method for the matrix polynomial case (3.2) is to reformulate the problem to a linear eigenproblem in \( jn \) dimensions provided either \( A_j \) or \( A_0 \) is nonsingular. Using an analogue of the companion matrix, problem (3.2) can be reformulated as

\[-A_0x_1 - A_1 x_2 - \ldots - A_{j-1} x_{j-1} = \lambda A_j x_j,\]

by setting \( x_1 = x \) and \( x_{i+1} = \lambda x_i \) for \( i = 1, \ldots, (j-1) \). By assuming \( w^T = (x_1^T, \ldots, x_j^T) \), then we can write

\[
M = \begin{pmatrix}
0 & 1 & 0 & \cdots & 0 \\
0 & 0 & 1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
A_0 & A_1 & \cdots & A_{j-1}
\end{pmatrix}
\quad \text{and} \quad
K = \begin{pmatrix}
1 & 0 & 0 & \cdots & 0 \\
0 & 1 & 0 & \cdots & 0 \\
0 & 0 & 1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \cdots & A_j
\end{pmatrix}.
\]

Equation (3.2) can be rewritten as \( Mw = \lambda Kw \) with \( M \) and \( K \) are \( jn \times jn \) matrices. The nonlinear eigenvalues of the original problem become the generalized eigenvalues of the pencil \([K, M]\). This companion matrix method is restricted to matrix polynomials. The methods of determinant evaluation and the companion matrix method were studied by Peters and Wilkinson [32].

The eigenproblem resulting from the exact modeling formulation is a nonlinear eigenproblem in the sense that the entries of the dynamic stiffness matrix are a nonlinear functions of the natural frequency. Of course, it is a special case from the general formula (3.1) if we define \( G(\lambda) \) as follows:

\[
G(\lambda) = K(\lambda) - \lambda M(\lambda) \tag{3.3}
\]

or

\[
G(\lambda) = K - \lambda M(\lambda).\]
3.2 Historical Background

The algorithms that solve the nonlinear eigenproblems may be divided into four categories:

1. Newton's techniques,
2. Inverse iteration methods,
3. Determinant search procedures,
4. Linearize, solve and then update approaches.

These methods may depend on evaluating derivatives or they may not use derivatives at all. In general, any technique solves these problems may take the following main steps.

1. Initialize: let $\lambda$ be an initial guess to the desired eigenvalue.
2. Solve: either solve a linear eigenvalue problem or factorize the matrix $G(\lambda)$.
3. Update: the eigenvalue $\lambda$.

The rest of this section is devoted to presenting at least one existing approach from each kind of the previous categories. We use the notations $G(\lambda_i) = G_i$ and $G'(\lambda_i) = G'_i$.

(1) Newton's Techniques

There are various forms of Newton's method [25, 37, 57] either by dealing with the problem as a system of nonlinear equations or by expanding $G(\lambda)$ via a Taylor series and then repeatedly solving the matrix pencil obtained from the first two terms of the Taylor series expansion of $G$ about $\lambda$, [37]. The general symmetric form of problem (3.1) can be converted to a system of $(n + 1)$ nonlinear equations

$$F \begin{pmatrix} x \\ \lambda \end{pmatrix} = \begin{pmatrix} G(\lambda)x \\ x^T x - 1 \end{pmatrix}$$

This system can be solved by using the following algorithm.
Algorithm 3.1

(1) Set: \( z_0 \) = an appropriate initial guess

(2) For \( i = 0, 1, 2, \cdots \) until convergence
   
   (a) Solve: \( F'_i \Delta z_i = -F_i \)
   
   (b) Compute: \( z_{i+1} = z_i + \Delta z_i \)
   
   (c) If, \( \| z_{i+1} - z_i \| \leq \epsilon \), then stop
   
   (d) \( z_i \leftarrow z_{i+1} \) go to 2

where,

\[
    z_i = \begin{pmatrix} x_i \\ \lambda_i \end{pmatrix} \quad \text{and} \quad F'_i = \begin{pmatrix} G_i & G'_i x \\ 2x^T & 0 \end{pmatrix}
\]

We can linearize problem (3.1) by using Taylor’s formula,

\[
    G_i(\lambda + \delta) = G(\lambda) + \delta G'(\lambda) + \frac{\delta^2}{2} R(\lambda, \delta)
\]

where, \( R(\lambda, \delta) \) is a matrix whose norm is bounded. Discarding \( R \), we can solve the problem by using the following algorithm:

Algorithm 3.2

(1) Choose: \( \lambda_0 \) = an appropriate initial guess

(2) For \( i = 0, 1, 2, \cdots \) until convergence
   
   (i) Solve: \(-G_i x_{i+1} = \mu_i G'_i x_{i+1} \)
   
   (ii) Set: \( \lambda_{i+1} = \lambda_i + \mu_i \)
   
   (iii) If, \( |\mu_i| \leq \epsilon \), then stop
   
   (iv) \( \lambda_i \leftarrow \lambda_{i+1} \) go to 2

If \( \mu_i \) is chosen as the absolutely smallest eigenvalue, then the convergence is quadratic. The convergence rate deteriorates when the problem is defective (i.e. does not have a
complete set of eigenvectors). The rate of convergence is linear for defective problems. Some drawbacks of these procedures are:

1. The derivative of the matrix $[G(\lambda)]$ may not exist or it may be hard to evaluate.
2. Newton's techniques are locally $q$-quadratic in the neighborhood of the solution, but not globally convergent in general. Thus, they may need another method for computing a good initial guess and they must be safeguarded.

(2) Inverse Iteration Methods

The idea of factorizing the matrix $[G(\lambda)]$ using permutation matrices and applying Newton–Raphson method for updating the parameter $\lambda$ has been proposed by V. N. Kublanovskaja (1969). The following are the main steps for this technique.

1. **Initialize**: the parameter, $\lambda$ by $\lambda_0$

2. **Factorize**: $G_iP_i = Q_iR_i$
   
   where $Q_i$ is unitary, usually a product of $n - 1$ elementary reflections, $R_i$ is an upper triangular matrix and $P_i$ is a permutation matrix chosen so that the diagonal entries of $R_i$ satisfy:
   
   $$|\rho_{11}| \geq |\rho_{22}| \geq \cdots \geq |\rho_{nn}|$$

3. **Set**: the function $\psi(\lambda)$ is set to be $\psi(\lambda) = \rho_{nn}$, where $\rho_{nn} = e_n^T R_i e_n$

4. **Update**: the parameter $\lambda$ by applying Newton–Raphson method on the function $\psi(\lambda)$.

M. I. Mavljanova [22] has studied the conditions for the quadratic convergence of this approach. The factorization in step (2) gives

$$y_i = Q_i e_n \quad \text{and} \quad x_i = P_i R_i^{-1} e_n$$

which are the approximations for the left and right eigenvectors, respectively. The eigenvector iteration starts a new in each step which makes this algorithm expensive.
Ruhe in [37] modified this algorithm by replacing the permutation matrix $P$ by orthonormal matrix $Q$. Some generalizations of inverse iteration are

**Subspace Iteration Methods**

Subspace iteration is a technique based on the inverse iteration and the Rayleigh–Ritz method in which several vectors are processed simultaneously. At each subspace iteration stage, the inverse iteration step supplies the admissible functions (which after each inverse iteration step become closer to the subspace being sought) and the Rayleigh–Ritz method provides the optimum combination of these admissible functions for use in the next round of the iteration. The Rayleigh–Ritz application also has the effect of orthogonalizing the trial vectors and hence helps to maintain linear independence at each iteration step. The following are the main steps for using this method for solving the nonlinear eigenproblems.

1. **Initialize**: the parameter, $\lambda$

2. **Construct** $Q_0 = [q_1, q_2, \cdots, q_k]$ be a set of linearly independent vectors in $\mathbb{R}^n$;

3. **Compute** $l$ orthogonal iterations:

   $$Q_i R_i = B(\lambda) Q_{i-1} \quad \text{where} \quad Q_i^T Q_i = I_k \quad \text{for} \quad i = 1, 2, \cdots, l$$

4. **Compute** the eigensolution $(\hat{\theta}_i^{(l)}, \hat{y}_i^{(l)})$ of the $k \times k$ matrix $A_k = Q_i^T B(\lambda) Q_i$ by the $QR$ algorithm and let

   $$F = [\hat{y}_1^{(l)}, \hat{y}_2^{(l)}, \cdots, \hat{y}_k^{(l)}]$$

5. **Test** the convergence of the required approximations $(\hat{\theta}_i^{(l)}, \hat{x}_i^{(l)})$ of $B(\lambda)$ where, $\hat{x}_i^{(l)} = Q_i \hat{y}_i^{(l)}$. If the convergence of the required approximations has not been achieved, then let $Q_0 = Q_l F$ and go to 2

6. **Update** the parameter $\lambda$

In 1991, Garratt used the subspace iteration method for solving the nonlinear eigenproblems with remarkable success [7]. The convergence of this method depends on
the ratio \( \frac{\lambda_p}{\lambda_{p+1}} \)(\( p > m \)), being small; thus if the eigenvalues are closely bunched, the process is necessarily slow in extracting the correct vectors. The method is best suited to a system in which the frequency spectrum has a large spread. Neumaier (1985), in [26], has proposed a technique for solving these problems which is

**Residual Inverse Iteration Method.**

This procedure can be described by the following five steps:

1. **Evaluate** an initial approximation \( v_0 \) to the eigenvector \( v \) as the normalized solution of the equation

\[
G(\sigma)\tilde{v}_0 = b, \quad v_0 = \tilde{v}_0/e^T\tilde{v}_0
\]

where, \( e \) is the unit vector with a 1 in the position of the largest entry of \( v \). The vector \( b \) is chosen to be a vector all its entries are ones.

2. **Compute** an improved approximation \( \lambda_{i+1} \) to the eigenvalue \( \lambda \) by solving the equation

\[
v_i^T G(\lambda_{i+1}) v_i = 0
\]

3. **Evaluate** the residual

\[
r_i = G(\lambda_{i+1}) v_i
\]

4. **Compute** an improved approximation \( v_{i+1} \) to the eigenvector \( v \) by solving the equation

\[
G(\sigma)\Delta v_i = r_i
\]

and normalizing the vector

\[
\tilde{v}_{i+1} = v_i - \Delta v_i, \quad v_{i+1} = \frac{v_i}{e^T\tilde{v}_{i+1}}
\]

5. **Increase** \( i \) by one and return to 2

This algorithm is equivalent to the ordinary inverse iteration in the linear case. If \( \sigma \) is sufficiently close to the smallest eigenvalue, convergence is at least linear with convergence factor proportional to \( |\sigma - \lambda| \). As with ordinary inverse iteration, the
convergence can be accelerated by using variable shifts.

(3) Determinant Search Techniques

This process simply requires the evaluation of $\det[G(\lambda)]$ for many values of $\lambda$ and observation of the sign changes in $\det[G(\lambda)]$. The value of the determinant can be calculated by factorizing the matrix $[G(\lambda)]$ to an upper triangular matrix by Gaussian elimination and then multiplying the elements on the diagonal of the tridiagonal matrix. If no pivoting is involved, the triangulation procedure will retain the banded form.

Bisection Method Combined With Sign Counting

The sign count is defined as the number of the negative elements on the leading diagonal of a matrix which has been triangulated by a simple Gaussian elimination process, in which the appropriate multiples of the pivotal row are added to unscaled succeeding rows without any row or column interchanges. The sign count method developed by Wittrick and Williams [56] is an infallible algorithm, based on the Sturm sequence, [10] which can for a trial value of $\lambda$, say $\lambda_t$, give the exact number of eigenvalues exceeded by $\lambda_1$. The algorithm in its full generality can be stated as

$$s[G(\lambda_t)] = J(\lambda_t) - J_0$$

where, $s[G(\lambda_t)]$ is the sign count of the upper triangular matrix that results from factorizing the matrix $[G(\lambda_t)]$ using Gaussian elimination process, $J(\lambda_t)$ is the integer number of eigenvalues exceeded by $\lambda_t$ and $J_0$ is the integer number of eigenvalues that would exceeded if constraints were imposed on the structure so as to make $x = 0$. Application of equation (3.4) leads to an effective method for convergence to any eigenvalue by "bisection" or "interval halving". Bisection initially involves the choice of two trial values of $\lambda$, say $\lambda_u$ and $\lambda_l$ such that $J(\lambda_u)$ exceeds the particular desired eigenvalue and $J(\lambda_l)$ does not. Further trial values of $\lambda$ can be then found by setting $\lambda_t = \frac{\lambda_l + \lambda_u}{2}$ where each new value of $\lambda_t$ will provide a different $\lambda_t$ or $\lambda_u$ depending on whether or not $J(\lambda_t)$ has exceeded the desired eigenvalue. By repeating the procedure,
any desired eigenvalue can be found to any required accuracy. If multiple eigenvalues of multiplicity \( m \) are encountered, they can also be located by bisection, since their presence will be detected by an increase of \( m \) in the sign count, \( J(\lambda_t) \), as \( \lambda_t \) crosses the \( m \)-fold repeated eigenvalue.

Simpson in [40] formulated an elegant method for locating the zeros of \( g_{nn} \) (the last element in the right corner from the upper triangular matrix that comes from Gaussian elimination process) by a quadratically convergent Newton–Raphson procedure in conjunction with the sign count algorithm. This method was introduced under the name

**A Newtonian Procedure Combined With Sign Counting.**

This technique can be summarized in the following steps:

(i) **Obtain** \( \lambda_0 \) as a first approximation by bisection as described in the previous algorithm.

(ii) **Form** \( G(\lambda_0) = E - \lambda_0 A \).

(iii) **Triangulate** \( G(\lambda_0) \) into an upper triangular matrix \( G_\Delta(\lambda_0) \) by simple Gaussian elimination.

(iv) **Set** \( x_n = 1 \) (the last component of the eigenvector \( x \)) and calculate \( x \) by a process of back substitution on \( G_\Delta(\lambda_0) \).

(v) **Find** The derivative of \( g_{nn} \) by using the scalar

\[
\frac{dg_{nn}}{d\lambda} = -x^T A x
\]

(vi) **Apply** Newton–Raphson method on the function \( g_{nn} \) to get \( \lambda_N \) in the form

\[
\lambda_N = \lambda_0 + \frac{g_{nn}}{x^T A x}
\]

(vii) **If** \( \lambda_N \) still resides between the poles bounding and the desired root, then set \( \lambda_0 = \lambda_N \) and reapply the algorithm from step (ii) until convergence is obtained. If \( \lambda_N \) does not reside between the pole bounds, then the procedure has to revert to the bisection step in (i) to obtain a new value of \( \lambda_0 \) which is closer to the desired root.
and the process from (ii) to (vii) is repeated.
The technique has certain deficiencies which affect its performance. Many triangulations may be required to locate any eigenvalue, and there is no indication of the position of the computed root in the frequency spectrum. Of course the bisection technique is safe and easy to implement but it is relatively slow owing to its linear convergence properties. Newton-Raphson may diverge if the starting point lies outside the neighborhood of the solution.

Intensive research has been conducted in this field by Swannell [51], Richards and Leung [36], Hallauer and Liu [11]. In [55] Williams and Kennedy give an improved zero-finding algorithm by combining parabolic interpolation with bisection method but it is slow for large problems. We are going to give a complete description of this method later on.

(4) Linearize, solve and then update technique
The main idea is to use an eigen-solver (k-step Arnoldi or IRL) to solve the linear eigenvalue subproblems that arise in conjunction with a zero-finding approach for updating the parameter $\lambda$. In (1992) H. A. Smith, D. C. Sorensen and R. Singh solved the dependent eigenproblems. They used IRL method to solve the linear eigenproblem and the secant method that based on the linear interpolation for updating the eigenvalue. This approach gives remarkable results but it still takes several iterations of the zero-finder to find a root.

3.3 MIRL Method
The Implicit Restarted Lanczos technique has been used with a considerable success to solve large scale algebraic eigenvalue problems resulting from composite membrane problems, discretization of convection-diffusion problems and linearized Navier Stokes problems up to dimension $10^4$ degrees-of-freedom on a variety of machines [46]. In [47], the mixed finite element model (1.21) has been compared with a procedure in [42] that proposed the eigensolution for a four-story frame example. The solutions of
this example is obtained experimentally and numerically by a variation of the exact model [41]. The authors in [47] deduced that:

1. The apparent difference between the experimental results and the analytical results is due to inaccuracies in modeling the exact material and the sectional properties of the test specimen.

2. The mixed model with an efficient technique for computing the nonlinear eigen-solutions gives superior results for higher and clustered eigenvalues.

3. The mixed model results are nearly exact when compared to the results reported in [42].

These advantages of IRL method and the structure of the nonlinear eigenvalue problems that come from the reformulated (mixed) finite element model indicate that the use of IRL technique to solve the linear eigenproblems will give accurate and efficient results. The main task of this thesis is to compute a few of the smallest eigenvalues $\lambda$ and their corresponding eigenvectors of the nonlinear eigenproblem

$$[K - \lambda M(\lambda)]x = 0, \quad \|x\| = 1$$

where, $K$ is banded symmetric positive definite matrix and $M(\lambda)$ is symmetric and its entries are transcendental functions of the the parameter $\lambda$. It is much better to cast this problem in the form

$$M(\lambda)x = \mu(\lambda)Kx, \quad \mu(\lambda) = \frac{1}{\lambda}.$$  \hspace{1cm} (3.5)

In this formulation Lanczos factorization becomes

$$M(\lambda)V = KVT + fe_k^T$$

with the columns of $V$ being $K$-orthogonal, and $T$ is tridiagonal. Let $\mu_1(\lambda)$ be the largest generalized eigenvalue of $T$ and let $y_1$ be the corresponding eigenvector. The
largest eigenvalue of $T$ approximates the largest eigenvalue of problem (3.5) at specific value of the parameter $\lambda$. This is typically, found very rapidly with the IRL iteration. A solution to (3.5) is obtained when we have solved the scalar equation

$$\mu_1(\lambda) = \frac{1}{\lambda} \text{ under the condition } \|f_\lambda\| \leq \varepsilon_1 \mu_1(\lambda)$$

where, $\mu_1(\lambda)$ and $y_1$ are the largest eigenvalue and its corresponding eigenvector of the matrix $T$, respectively. A variety of standard and specialized numerical techniques (such as the bisection technique, the Newton-Raphson method and the secant approach) may be applied to this scalar problem and we discuss some of these techniques in the next chapter. In the implementation of the Lanczos method (Algorithm 2.1) the only way that the symmetric matrix $A$ is involved in the computation is during step (3), where the product $Av$ needs to be formed. To do this economically requires a procedure, which takes full advantage of the sparsity and the symmetry of $K$ and $M(\lambda)$. It does not require an explicit formulation of $A$. The product $A(\lambda)v_k = L^{-1}M(\lambda)L^{-T}v_k$ can be computed without additional cost by factorizing the matrix $K$ only one time. The factorization is subsequently used repeatedly to compute this matrix vector product at each $\lambda$. Assume that the vectors $z$ and $u$ are temporary $n \times 1$ vector arrays and the matrix $K$ has been factored to $LL^T$ using the Cholesky factorization method. The following algorithm is used for computing the product $A(\lambda)v_k$.

**Algorithm 3.3**

function $[w] = \text{product}(L, M(\lambda), v_k)$

(a) **Solve**: $L^Tv = u$ for $u$ by a back substitution;

(b) **Form**: $z = M(\lambda)u$

(c) **Solve**: $Lw = z$ for $w$ by a forward substitution.

Instead of solving problem (3.5) the transformed problem

$$A(\lambda)x = \mu(\lambda)x$$

(3.6)
will be solved. The problem becomes standard eigenvalue problem at specific values of \( \lambda \) and \( A(\lambda) \) is still symmetric. The Lanczos method is determined such that step (3) of algorithm 2.1 is replaced by algorithm 3.3, i.e \( w = A(\lambda)v_k \). The IRL technique (algorithm 2.2) becomes

\[
\text{function}[T, V, f] = \text{MIRL}(A, L, k, p, \text{tol})
\]

(3.7)

with the same steps but it calls algorithm 2.1 which calls algorithm 3.3 instead of step (3).

The entries of the matrix \( A(\lambda) \) are ratios of transcendental functions. The generalized eigenvalues of the matrix \( A(\lambda) \) at specific values of \( \lambda \), \( \mu(\lambda) \) are the roots of a characteristic polynomial that has rational coefficients. The relationship between the roots and the coefficients of the polynomials lead us to approximate the generalized eigenvalues \( \mu(\lambda) \), with rational functions. The MIRL method can be applied to evaluate a subset of the largest generalized eigenvalues \( \mu(\lambda) \) for the linear eigenvalue subproblems that appear during the solution of problem (3.6). A scalar zero-finder based on rational interpolation can be applied for updating the eigenvalue \( \lambda \). The solution of these problems is the intersection point between \( \frac{1}{\lambda} \)-curve and \( \mu(\lambda) \)-curve as shown in figures 3.1 and 3.2.

Figure 3.1 illustrates the behavior of the generalized eigenvalues \( \mu(\lambda) \) and the characterization of the solution of the nonlinear eigenproblem that is represented by the intersection points between the \( \mu \)-curves and the \( \frac{1}{\lambda} \)-curve. Each \( \mu \)-curve represents the generalized eigenvalues for a symmetric tridiagonal matrix of order \( 50 \times 50 \). The entries of this matrix are \( \lambda^3 + 2\lambda^2 - 1 \) on the main diagonal and \( \lambda \sin(\lambda) - \lambda^2 + \lambda \) on the subdiagonals that are evaluated at a specific value for the parameter \( \lambda \). Figure 3.2 graphically depicts the tenth smallest eigenvalue for the nonlinear problem as the intersection between the \( \mu \)-curve and the \( \frac{1}{\lambda} \)-curve.

Remark: These graphs represent the spectrum of a potential matrix. This problem is not a structure example, but its entries are nonlinear functions.
Figure 3.1  The description of the nonlinear eigenproblem solution

Figure 3.2  The tenth smallest eigenvalue for a nonlinear problem
Chapter 4

THE MONOTONICITY ANALYSIS THEORY

The monotonicity analysis plays an important role in the existence and uniqueness of the solution for both linear and nonlinear systems. The starting point in this relationship is to consider how we might go about minimizing the functional $\psi(x)$ defined by

$$
\psi(x) = \frac{1}{2} x^T M x - x^T b
$$

where $b \in \mathbb{R}^n$ and $M \in \mathbb{R}^{n \times n}$. Minimizing the quadratic (4.1) and solving the linear system $Mx = b$ are equivalent problems. This linear system has a solution when $M$ is positive semi-definite and this solution is unique if $M$ is strictly positive definite. On the other hand, the quadratic (4.1) has a solution when the functional $\psi$ is convex (definition 1.1) and this solution is unique when this quadratic is strictly convex.

In 1992, H. A. Smith, D. C. Sorensen and R. K. Singh have said, in [47], "we have observed interesting behavior of the function $\mu_j(\lambda)$ associated with these frame-type structures. It appears that these functions are both increasing and convex in an interval $[0, u]$ containing the solution $\lambda_j$. We have not been able to verify this theoretically." This chapter analyzes the behavior of the functions $\mu_j(\lambda)$. Our analysis is sufficiently general that it holds for any system that can be formulated by using the mixed finite element technique for the axial vibration. The first and second parts are concerned with establishing lemmas that lead to the proof of the positive definiteness for the global mass matrices in the mixed and the exact finite element models. In addition, the positive definiteness of the stiffness matrix associated with the exact finite element model in a specific interval is proved. The last section is devoted to
studying the monotonicity analysis of the functions $\mu_j(\lambda)$ which are the generalized eigenvalues of the matrices $L^{-1}M(\lambda)L^{-T}$ at specific values of the parameter $\lambda$.

### 4.1 Positive Definiteness for the Mixed Model

The mixed finite element model (1.21) results from globalizing the elemental mass matrix, (1.20) and the elemental stiffness matrix, $k_p$ for the axial vibration. The entries of the elemental mass matrix can be expanded in Maclaurin’s series about the origin in terms of $\bar{\nu}, \tilde{\nu} = \nu l \in \mathcal{R}$, where $\mathcal{R}$ is the set of all nonnegative real numbers.

The expansion of the sine function can be written as

$$
\tilde{\nu} \sin \bar{\nu} = \tilde{\nu}^2 \left(1 - \frac{\tilde{\nu}^2}{3!} + \frac{\tilde{\nu}^4}{5!} - \frac{\tilde{\nu}^6}{7!} + \frac{\tilde{\nu}^8}{9!} - O(\tilde{\nu}^{10}) \right).
$$

Inverting both sides of the previous equality we get

$$
(\tilde{\nu} \sin \bar{\nu})^{-1} = \frac{1}{\tilde{\nu}^2} \left(1 - \left(\frac{\tilde{\nu}^2}{3!} - \frac{\tilde{\nu}^4}{5!} + \frac{\tilde{\nu}^6}{7!} - \frac{\tilde{\nu}^8}{9!} + O(\tilde{\nu}^{10})\right)^{-1}.
$$

The expansion of the right hand side of the last equality using binomial theorem gives

$$
(\tilde{\nu} \sin \bar{\nu})^{-1} = \frac{1}{\tilde{\nu}^2} \left(1 + \frac{1}{3!} \tilde{\nu}^2 + \frac{7}{3(5!)} \tilde{\nu}^4 + \frac{31}{3(7!)} \tilde{\nu}^6 + \frac{381}{5(9!)} \tilde{\nu}^8 + O(\tilde{\nu}^{10})\right).
$$

Using Maclaurin’s expansion of the cosine function, equation (4.3) and rearranging terms give

$$
\cos \tilde{\nu}(\tilde{\nu} \sin \bar{\nu})^{-1} = \frac{1}{\tilde{\nu}^2} \left(1 - \frac{2}{3!} \tilde{\nu}^2 - \frac{8}{3(5!)} \tilde{\nu}^4 - \frac{32}{3(7!)} \tilde{\nu}^6 - \frac{1812}{5(9!)} \tilde{\nu}^8 - O(\tilde{\nu}^{10})\right).
$$

From equations (4.3) and (4.4), we get

$$
\frac{1}{\tilde{\nu}^2} - \cos \tilde{\nu}(\tilde{\nu} \sin \bar{\nu})^{-1} = \frac{2}{3!} + \frac{8}{3(5!)} \tilde{\nu}^2 + \frac{32}{3(7!)} \tilde{\nu}^4 + \frac{1812}{5(9!)} \tilde{\nu}^6 + O(\tilde{\nu}^8),
$$

$$
(\tilde{\nu} \sin \bar{\nu})^{-1} - \frac{1}{\tilde{\nu}^2} = \frac{1}{3!} + \frac{7}{3(5!)} \tilde{\nu}^2 + \frac{31}{3(7!)} \tilde{\nu}^4 + \frac{381}{5(9!)} \tilde{\nu}^6 + O(\tilde{\nu}^8).
$$
By substituting from equations (4.5) and (4.6) into the matrix (1.20), we get

\[ m_x(\bar{\nu}) = \hat{m} l \begin{pmatrix}
\frac{2}{3!} + \frac{8}{3(5!)} \bar{\nu}^2 + \frac{32}{3(7!)} \bar{\nu}^4 + \frac{1812}{5(9!)} \bar{\nu}^6 \\
\frac{1}{3!} + \frac{7}{3(5!)} \bar{\nu}^2 + \frac{31}{3(7!)} \bar{\nu}^4 + \frac{381}{5(9!)} \bar{\nu}^6
\end{pmatrix} + O(\bar{\nu}^8) \]  

(4.7)

**Remark**: The terms of the expansions (4.5) and (4.6) are positive for any real value \( \bar{\nu} \). The coefficients of the expansion (4.5) are always larger than the corresponding coefficients of expansion (4.6).

**Lemma 4.1** The matrix \( m_x(\bar{\nu}) \) is positive definite for all values of \( \bar{\nu} \).

**Proof**: The difference between the two entries in the first row of the matrix (4.7) is

\[ \hat{m} l \left( \frac{1}{3!} + \frac{1}{3(5!)} \bar{\nu}^2 + \frac{1}{3(7!)} \bar{\nu}^4 + \frac{1431}{5(9!)} \bar{\nu}^6 + O(\bar{\nu}^8) \right) \]  

(4.8)

which is positive for any value of \( \bar{\nu} \). The addition of these two entries is

\[ \hat{m} l \left( \frac{1}{2} + \frac{1}{4!} \bar{\nu}^2 + \frac{3}{6!} \bar{\nu}^4 + \frac{2193}{5(9!)} \bar{\nu}^6 + O(\bar{\nu}^8) \right) \]  

(4.9)

which is positive for all values of \( \bar{\nu} \). The principal minors of the matrix, \([m_x(\bar{\nu})]\) are positive because \( \text{det}[m_x(\bar{\nu})](>0) \) equals the product of the two terms (4.8) and (4.9) and the first entry of this matrix is positive for any value of \( \bar{\nu} \), which completes the proof by using theorem 1.1.

**Corollary**: In the static case (\( \bar{\nu} = 0 \)) the matrix \( m_x(\bar{\nu}) \) takes the form

\[ m_x(0) = \frac{\hat{m} l}{6} \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} \]

which is the same matrix that we had before by using polynomial displacements shape functions.

**Lemma 4.2** The global mass matrix \([M(\bar{\nu})]\) is symmetric positive definite.
Proof: This global mass matrix results from applying the assembly process on the
elemental mass matrix \([m_x(\bar{v})]\) followed by eliminating the rows and columns corresponding to the boundary conditions. The entries of this matrix, \([M(\bar{v})]\) are:

\[
2\hat{m}l\left(\frac{2}{3!} + \frac{8}{3(5!)}\bar{v}^2 + \frac{32}{3(7!)}\bar{v}^4 + \frac{1812}{5(9!)}\bar{v}^6 + O(\bar{v}^8)\right)
\]
on the main diagonal. Each entry on the subdiagonal and superdiagonal is

\[
\hat{m}l\left(\frac{1}{3!} + \frac{7}{3(5!)}\bar{v}^2 + \frac{31}{3(7!)}\bar{v}^4 + \frac{381}{5(9!)}\bar{v}^6 + O(\bar{v}^8)\right).
\]

This matrix is strictly diagonally dominant and its entries are positive for all \(\bar{v}\). The result is established by using lemma 1.3. □

This proof is demonstrated graphically in figure 4.1

Theorem 4.1 If \(L\) is a unit triangular matrix resulting from decomposing the global stiffness matrix \(K\) into \(LL^T\), then the matrix \(L^{-1}M(\bar{v})L^{-T}\) is positive definite for all values of \(\bar{v}\), where \(A(\bar{v}) = L^{-1}M(\bar{v})L^{-T}\).

The proof is straight forward from lemma 4.2 and lemma 1.2. Figure 4.3 illustrates the positive definiteness property for this type of matrices

4.2 Positive Definiteness for the Exact Model

The exact finite element model is assembled by globalizing the elemental mass, (1.16) and the elemental stiffness, (1.17) matrices for the axial vibration. The entries of these elemental mass and stiffness matrices can be expanded in Maclaurin’s series about the origin and some terms in \(\bar{v}\). From equality (4.2) we get

\[
\sin \bar{v} = \bar{v}(1 - \frac{\bar{v}^2}{3!} + \frac{\bar{v}^4}{5!} - \frac{\bar{v}^6}{7!} + \frac{\bar{v}^8}{9!} - O(\bar{v}^{10})).
\]

Then we can write

\[
(\sin \bar{v})^{-2} = \frac{1}{\bar{v}^2}(1 - \frac{\bar{v}^2}{3!} - \frac{\bar{v}^4}{5!} + \frac{\bar{v}^6}{7!} - \frac{\bar{v}^8}{9!} + O(\bar{v}^{10}))^{-2}.
\]
Using the binomial theorem for the right hand side from the previous formula we get

\[(\sin \tilde{\nu})^{-2} = \frac{1}{\tilde{\nu}^2}(1 + \frac{2}{3!} \tilde{\nu}^2 + \frac{24}{3(5!)} \tilde{\nu}^4 + \frac{160}{3(7!)} \tilde{\nu}^6 + \frac{2688}{5(9!)} \tilde{\nu}^8 + O(\tilde{\nu}^{10})). \tag{4.10} \]

Multiplying the previous equality by Maclaurin's expansion for the cosine gives

\[\cos \tilde{\nu}(\sin \tilde{\nu})^{-2} = \frac{1}{\tilde{\nu}^2}(1 - \frac{1}{3!} \tilde{\nu}^2 - \frac{21}{3(5!)} \tilde{\nu}^4 - \frac{155}{3(7!)} \tilde{\nu}^6 - \frac{2667}{5(9!)} \tilde{\nu}^8 + O(\tilde{\nu}^{10})). \tag{4.11} \]

From equations (4.10) and (4.4) we get

\[\frac{1}{(\sin \tilde{\nu})^2} - \frac{\cos \tilde{\nu}}{\tilde{\nu} \sin \tilde{\nu}} = \frac{4}{3!} + \frac{32}{3(5!)} \tilde{\nu}^2 + \frac{192}{3(7!)} \tilde{\nu}^4 + \frac{4332}{5(9!)} \tilde{\nu}^6 + O(\tilde{\nu}^8). \tag{4.12} \]

Subtracting equality (4.11) from equality (4.3) gives

\[\frac{1}{\tilde{\nu} \sin \tilde{\nu}} - \frac{\cos \tilde{\nu}}{(\sin \tilde{\nu})^2} = \frac{2}{3!} + \frac{28}{3(5!)} \tilde{\nu}^2 + \frac{186}{3(7!)} \tilde{\nu}^4 + \frac{3048}{5(9!)} \tilde{\nu}^6 + O(\tilde{\nu}^8). \tag{4.13} \]

Introducing equations (4.12) and (4.13) into matrix (1.16), the elemental mass matrix can be written as.

\[m_{e}(\tilde{\nu}) = \frac{\hat{m}_l}{2} \left( \begin{array}{cccc}
\frac{4}{3!} & \frac{32}{3(5!)} & \frac{192}{3(7!)} & \frac{4332}{5(9!)} \\
\frac{2}{3!} & \frac{28}{3(5!)} & \frac{186}{3(7!)} & \frac{3048}{5(9!)} \\
\frac{2}{3!} & \frac{28}{3(5!)} & \frac{186}{3(7!)} & \frac{3048}{5(9!)} \\
\frac{2}{3!} & \frac{28}{3(5!)} & \frac{186}{3(7!)} & \frac{3048}{5(9!)} \\
\end{array} \right) + O(\tilde{\nu}^8). \tag{4.14} \]

**Lemma 4.3** The matrix \(m_e(\tilde{\nu})\) is positive definite for all values of \(\tilde{\nu}\).

**Proof**: The first entry of this matrix is positive for any value of \(\tilde{\nu}\). The difference between the two entries in the first row is

\[\hat{m}_l(\frac{1}{3!} + \frac{2}{3(5!)} \tilde{\nu}^2 + \frac{1}{7!} \tilde{\nu}^4 + \frac{642}{5(9!)} \tilde{\nu}^6 + O(\tilde{\nu}^8))\]

which is positive for all \(\tilde{\nu}\). The addition of these two entries is

\[\hat{m}_l(\frac{1}{2} + \frac{2}{4!} \tilde{\nu}^2 + \frac{189}{7!} \tilde{\nu}^4 + \frac{738}{9!} \tilde{\nu}^6 + O(\tilde{\nu}^8))\]
which is also positive. We deduce that \( \det[m_\epsilon(\bar{\nu})] > 0 \). Thus all the principal minors of \( [m_\epsilon(\bar{\nu})] \) are positive and the proof is completed by theorem 1.1

**Corollary:** In the static case \((\bar{\nu} = 0)\) the matrix \( m_\epsilon(\bar{\nu}) \) takes the form

\[
m_\epsilon(0) = \frac{\bar{\nu} l}{6} \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}
\]

which is the same matrix that we obtained before by using polynomial displacements shape functions

**Lemma 4.4** The global mass matrix \( M(\bar{\nu}) \) is positive definite for all values of \( \bar{\nu} \).

**Proof:** After applying the assembly process on the symmetric positive definite (lemma 4.3), elemental mass matrix \( m_\epsilon(\bar{\nu}) \) and eliminating the first row and its corresponding column (due to the boundary conditions), the entries of the global matrix \( [M(\bar{\nu})] \) are:

\[
\alpha_{ij} = \begin{cases} 
2 \left( \frac{4}{3!} + \frac{32}{3!(5!)} \bar{\nu}^2 + \frac{192}{3!(7!)} \bar{\nu}^4 + \frac{4332}{5!(9!)} \bar{\nu}^6 + O(\bar{\nu}^8) \right), & \text{if } i = j \text{ and } i, j = 1, 2, \ldots, n-1 \\
\frac{2}{3!} + \frac{28}{3!(5!)} \bar{\nu}^2 + \frac{186}{3!(7!)} \bar{\nu}^4 + \frac{3046}{5!(9!)} \bar{\nu}^6 + O(\bar{\nu}^8), & \text{if } |i - j| = 1 \\
0, & \text{elsewhere.}
\end{cases}
\]

Since, \(|\alpha_{ii}| > |\alpha_{ij}| + |\alpha_{ji}| \) and \( j = i + 1 \), the global matrix is strictly diagonally dominant with positive entries. Using lemma 1.3, the matrix \([M(\bar{\nu})]\) is positive definite for all values of \( \bar{\nu} \).

**Figure 4.2** graphically emphasizes the result of this lemma.

Using equations (4.10), (4.3), (4.4) and (4.11), we have

\[
\bar{\nu}^2 \sin \bar{\nu} = 1 + \frac{2}{3!} \bar{\nu}^2 + \frac{24}{3!(5!)} \bar{\nu}^4 + \frac{160}{3!(7!)} \bar{\nu}^6 + \frac{2688}{5!(9!)} \bar{\nu}^8 + O(\bar{\nu}^{10}),
\]

\[
\bar{\nu} \sin \bar{\nu}^{-1} = 1 + \frac{1}{3!} \bar{\nu}^2 + \frac{7}{3!(5!)} \bar{\nu}^4 + \frac{31}{3!(7!)} \bar{\nu}^6 + \frac{381}{5!(9!)} \bar{\nu}^8 + O(\bar{\nu}^{10}),
\]

\[
\bar{\nu} \cos \nu \sin \nu^{-1} = 1 - \frac{2}{3!} \bar{\nu}^2 - \frac{8}{3!(5!)} \bar{\nu}^4 - \frac{32}{3!(7!)} \bar{\nu}^6 - \frac{1644}{5!(9!)} \bar{\nu}^8 - O(\bar{\nu}^{10}),
\]

\[
\bar{\nu}^2 \cos \nu \sin \nu^{-2} = 1 - \frac{1}{3!} \bar{\nu}^2 - \frac{21}{3!(5!)} \bar{\nu}^4 - \frac{155}{3!(7!)} \bar{\nu}^6 - \frac{2667}{5!(9!)} \bar{\nu}^8 - O(\bar{\nu}^{10}).
\]
By using the previous equations we get

\[ \ddot{\nu}^2 \sin^2 \nu + \dot{\nu} \cos \nu \sin \nu = \frac{16}{3(5!)} \nu^4 + \frac{128}{3(7!)} \nu^6 + \frac{1044}{5(9!)} \nu^8 + O(\nu^{10}), \quad (4.15) \]

\[ -(\dot{\nu} \sin \nu)^{-1} + \nu^2 \cos \nu \sin \nu + O(\nu^{10}) \quad (4.16) \]

Introducing equations (4.15), (4.16) into the matrix (1.17), the elemental stiffness matrix becomes

\[ k_e(\nu) = \frac{\dot{\nu}}{2l} \begin{pmatrix} 2 & \frac{16}{3(5!)} \nu^4 + \frac{128}{3(7!)} \nu^6 + \frac{1044}{5(9!)} \nu^8 & -2 & \frac{14}{3(5!)} \nu^4 + \frac{124}{3(7!)} \nu^6 + \frac{2286}{5(9!)} \nu^8 \\ -2 & \frac{14}{3(5!)} \nu^4 + \frac{124}{3(7!)} \nu^6 + \frac{2286}{5(9!)} \nu^8 & 2 & \frac{16}{3(5!)} \nu^4 + \frac{128}{3(7!)} \nu^6 + \frac{1044}{5(9!)} \nu^8 \end{pmatrix} \quad (4.17) \]

Remark: The terms of the series (4.15) and (4.16) are even-power expansions with positive coefficients which are strictly decreasing. The coefficients of the series (4.16) which are starting from coefficient of \( \nu^8 \) are larger than their correspondent of the series (4.15). The difference between the two entries in the first row of the matrix \( k_e(\nu) \) is \( \leq 0 \) at \( \nu = 3 \).

**Lemma 4.5** The matrix \( k_e(\nu) \) is positive definite for all values of \( \nu \), such that \( \nu \in (0, 3) \).

**Proof**: Suppose that

\[ \alpha = \frac{\dot{\nu}}{l} \left( 1 + \frac{8}{3(5!)} \nu^4 + \frac{64}{3(7!)} \nu^6 + \frac{522}{5(9!)} \nu^8 + O(\nu^{10}) \right), \]

\[ \beta = \frac{\dot{\nu}}{l} \left( -1 + \frac{7}{3(5!)} \nu^4 + \frac{62}{3(7!)} \nu^6 + \frac{1143}{5(9!)} \nu^8 + O(\nu^{10}) \right). \]

The entry, \( \alpha \) of this matrix is positive for any value of \( \nu \). Since

\[ \alpha + \beta = \frac{\dot{\nu}}{l} \left( \frac{15}{3(5!)} \nu^4 + \frac{126}{3(7!)} \nu^6 + \frac{1665}{5(9!)} \nu^8 + O(\nu^{10}) \right), \quad (4.18) \]
\[ \alpha - \beta = \frac{\hat{A}E}{l} \left( 2 + \frac{1}{3(5!)} \bar{\nu}^4 + \frac{2}{3(7!)} \bar{\nu}^6 - \frac{621}{5(9!)} \bar{\nu}^8 + O(\bar{\nu}^{10}) \right). \] (4.19)

These two terms (4.18) and (4.19) are positive, but \[ \det[k_\nu(\bar{\nu})] = \alpha^2 - \beta^2 > 0 \] for \( \bar{\nu} \in (0, 3) \). Thus, the principal minors of the matrix \([k_\nu(\bar{\nu})]\) are positive and the proof is completed by theorem 1.1.

**Corollary:** In the static case \( (\bar{\nu} = 0) \) the matrix \( k_\nu(\bar{\nu}) \) takes the form

\[ k_\nu(0) = \frac{\hat{A}E}{l} \begin{pmatrix} 1 & -1 \\ 0 & 1 \end{pmatrix} \]

which is the same matrix that we got before by using polynomial displacements shape functions.

**Lemma 4.6** The global matrix \( K(\bar{\nu}) \) is positive definite for all values of \( \bar{\nu} \), such that \( \bar{\nu} \in (0, 3) \).

**Proof:** The matrix, \([K(\bar{\nu})]\) is the resulting matrix from applying the assembly process on the symmetric positive definite (lemma 4.5) elemental stiffness matrix \([k_\nu(\nu)]\), then eliminate the columns and the rows that are corresponding to the boundary conditions. The entries of \([K(\bar{\nu})]\) are:

\[
2 \left( 2 + \frac{16}{3(5!)} \bar{\nu}^4 + \frac{128}{3(7!)} \bar{\nu}^6 + \frac{1044}{5(9!)} \bar{\nu}^8 + O(\bar{\nu}^{10}) \right) \text{ and } -2 \left( 2 + \frac{14}{3(5!)} \bar{\nu}^4 + \frac{124}{3(7!)} \bar{\nu}^6 + \frac{2286}{5(9!)} \bar{\nu}^8 + O(\bar{\nu}^{10}) \right)
\]

on the main diagonal and subdiagonal (and superdiagonal), respectively. This global stiffness matrix is strictly diagonally dominant with positive entries on the interval, \( (0, 3) \). Using lemma 1.3 then \( K(\bar{\nu}) \) is positive definite for all \( \bar{\nu} \in [0, 3) \).

This positive definite behavior of the global stiffness matrix is assured graphically by figure 4.4.

**Theorem 4.2** If \( L(\bar{\nu}) \) is a unit triangular matrix resulting from decomposing the global stiffness matrix \( K(\bar{\nu}) \) into \( L(\bar{\nu})L(\bar{\nu})^T \) for all values of \( \bar{\nu} \in [0, 3) \), then the matrix \( L^{-1}(\bar{\nu})M(\bar{\nu})L^{-T}(\bar{\nu}) \) is symmetric positive definite.
The proof follows from the definition of $M(\bar{\nu})$ and $K(\bar{\nu})$ and lemmas 4.4 and 1.2. Figure 4.5 shows this behavior.

### 4.3 The Monotonicity Analysis

The new technique developed in this thesis is designed to solve the nonlinear eigenvalue problem (3.4) that may be rewritten as

$$A(\lambda)x = \mu(\lambda)x, \quad A(\lambda) = L^{-1}M(\lambda)L^{-T},$$

where $L$ is a unit lower triangular matrix, $M(\lambda)$ is symmetric positive definite for any specific value of $\lambda$ and $\mu(\lambda) = \frac{1}{\lambda}$. The main idea of this procedure is the well known technique "linearize, solve and then update". This idea is to solve a linear eigenvalue problem at a specific value of the parameter $\lambda$, say $\hat{\lambda}_j$, and approximate the generalized eigenvalue $\mu_i(\hat{\lambda}_j)$. The value $\hat{\lambda}_j$ is updated by solving the scalar equation $\lambda\mu(\lambda) = 1$. The question now is what is the behavior of the generalized eigenvalues $\mu^{(j)}_i$ where, $\mu^{(j)}_i = \mu_i(\hat{\lambda}_j)$, $j = 1, 2, \cdots$?

**Lemma 4.7** The matrix $[m_x(\bar{\nu}_2) - m_x(\bar{\nu}_1)]$ is symmetric positive definite for all $\bar{\nu}_2 > \bar{\nu}_1$.

**Proof** The symmetry of the matrix $[m_x(\bar{\nu}_2) - m_x(\bar{\nu}_1)]$ is inherited from the symmetry of the matrix (4.7), for any value of $\bar{\nu}$. The entries of the matrix $[m_x(\bar{\nu}_2) - m_x(\bar{\nu}_1)]$ are

$$\alpha_{11} = \hat{m}l \left( \frac{8}{3(5!)}(\bar{\nu}_2^2 - \bar{\nu}_1^2) + \frac{32}{7!}(\bar{\nu}_2^4 - \bar{\nu}_1^4) + \frac{1812}{5(9!)}(\bar{\nu}_2^6 - \bar{\nu}_1^6) + O(\nu^8) \right),$$

$$\alpha_{12} = \hat{m}l \left( \frac{7}{3(5!)}(\bar{\nu}_2^2 - \bar{\nu}_1^2) + \frac{31}{3(7!)}(\bar{\nu}_2^4 - \bar{\nu}_1^4) + \frac{381}{5(9!)}(\bar{\nu}_2^6 - \bar{\nu}_1^6) + O(\nu^8) \right),$$

$$\alpha_{21} = \alpha_{12},$$

$$\alpha_{22} = \alpha_{11}.$$

Note $\bar{\nu}_2 > \bar{\nu}_1$ implies $\alpha_{11} > 0$ and $\alpha_{11} - \alpha_{12} > 0$. This indicates that all the principal minors of the matrix $[m_x(\bar{\nu}_2) - m_x(\bar{\nu}_1)]$ are positive. The proof is completed by using theorem 1.1. □
Lemma 4.8  The global matrix \([M(\tilde{v}_2) - M(\tilde{v}_1)]\) is symmetric positive definite for all \(\tilde{v}_2 > \tilde{v}_1\).

Proof: The matrix \([M(\tilde{v}_2) - M(\tilde{v}_1)]\) results from applying the assembly process on the symmetric positive definite elemental matrix \([m_x(\tilde{v}_2) - m_x(\tilde{v}_1)]\) and eliminating the rows and columns corresponding to the boundary conditions. The entries of this global matrix are \(\alpha\) on the main diagonal and each entry on the subdiagonal and on the superdiagonal is \(\beta\) where,

\[
\begin{align*}
\alpha &= 2\tilde{m}l \left( \frac{8}{3(5!)}(\tilde{v}_2^2 - \tilde{v}_1^2) + \frac{32}{7!}(\tilde{v}_2^4 - \tilde{v}_1^4) + \frac{1812}{5(9!)}(\tilde{v}_2^6 - \tilde{v}_1^6) + O(\tilde{v}^8) \right) \text{ and} \\
\beta &= \tilde{m}l \left( \frac{7}{3(5!)}(\tilde{v}_2^2 - \tilde{v}_1^2) + \frac{31}{3(7!)}(\tilde{v}_2^4 - \tilde{v}_1^4) + \frac{381}{5(9!)}(\tilde{v}_2^6 - \tilde{v}_1^6) + O(\tilde{v}^8) \right).
\end{align*}
\]

The entries \(\alpha\) and \(\beta\) are positive for any real value \(\tilde{v}\) and \(\alpha > 2\beta\). The matrix \([M(\tilde{v}_2) - M(\tilde{v}_1)]\) is strictly diagonally dominant and the proof is completed by lemma 1.3.

Corollary: From theorem 4.1 the matrices \(A(\tilde{v}_2)\) and \(A(\tilde{v}_1)\) are symmetric positive definite and it follows directly from lemma 4.8 that the matrix \([A(\tilde{v}_2) - A(\tilde{v}_1)]\) is positive definite.

Theorem 4.3 Suppose \(\tilde{v}_2 > \tilde{v}_1\) which implies \([A(\tilde{v}_2) - A(\tilde{v}_1)]\) is positive definite. Let the eigenvalues of \([A(\tilde{v})]\) are ordered such that \(\mu_1(\tilde{v}) \geq \mu_2(\tilde{v}) \geq \cdots \geq \mu_n(\tilde{v})\), then \(\mu_j(\tilde{v})\) is an increasing function of \(\tilde{v}\) for each \(j\), where \(j = 1, 2, \ldots, n\).

Proof: Suppose \(\tilde{v}_2 > \tilde{v}_1\). \(\mathcal{R}\) is any subspace of \(\mathbb{R}^n\) and \(q(\neq 0) \in \mathcal{R}\) such that \(\|q\|_2 = 1\). Applying the minmax theorem 1.4, we get

\[
\mu_j(\tilde{v}_2) = \max_{\dim\mathcal{R} = j} \{ \min_{q \in \mathcal{R}} (q^T A(\tilde{v}_2)q) \} \\
= \max_{\dim\mathcal{R} = j} \{ \min_{q \in \mathcal{R}} (q^T A(\tilde{v}_1)q + q^T [A(\tilde{v}_2) - A(\tilde{v}_1)]q) \} \\
\geq \max_{\dim\mathcal{R} = j} \{ \min_{q \in \mathcal{R}} (q^T A(\tilde{v}_1)q) \} \\
= \mu_j(\tilde{v}_1).
\]
This completes the proof of the monotone increasing property for the generalized eigenvalues $\mu_j(\bar{\nu})$ for each $j$. ■

The continuity and the differentiability of these functions, $\mu_j(\bar{\nu})$ are inherited from the entries of the elemental mass matrix $m_x(\bar{\nu})$ which are continuously differentiable on $R^+$ (transcendental functions), where $R^+$ is the set of all real positive numbers.

The following graphs show the behavior of the spectrum of the mass and stiffness matrices in both mixed and exact finite element models. We globalize the elemental mass matrix for the mixed finite element model, (4.7), and the elemental mass and stiffness matrices for the exact finite element model (4.14) and (4.17), respectively. This globalization is assembled by using the assembly process of an axial element. For element $i$ with nodes $i$ and $i+1$, the two rows and columns of the elemental matrix are added into rows and columns $i$ and $i+1$ of the matrix for the complete system. After the final stage of the assembly process, we eliminate the first row and the first column from the global matrix. Figures 4.1, 4.2 and 4.4 are generated by taking 50 values for the parameter $\lambda$ and plotting the spectrum of each resulting matrix at a specific value of $\lambda$. These figures represent the spectrum for the global mixed and exact mass matrices and global exact stiffness matrix, respectively. Figures 4.3 and 4.5 represent the spectrum for the matrices $L^{-1}M(\lambda)L^{-T}$ and $L^{-1}(\lambda)M(\lambda)L^{-T}(\lambda)$, respectively. We used the same assembly process to get the global matrices $M(\lambda)$ and $K = LL^T$. 
The generalized eigenvalue curves

Figure 4.1 The spectrum of the mass matrix associated with the mixed finite element model.

The generalized eigenvalue curves

Figure 4.2 The spectrum of the mass matrix associated with the exact finite element model.
Figure 4.3 The spectrum of the matrix $A(\lambda)$. 
Figure 4.4 The spectrum of stiffness matrix associated with the exact finite element model.

Figure 4.5 The spectrum of the matrix $L^{-1}(\lambda)M(\lambda)L^{-T}(\lambda)$. 
Chapter 5

SOME ZERO-FINDING TECHNIQUES

The root-finding approaches that are used for updating the parameter $\lambda$ in the symmetric nonlinear eigenvalue problems may be regarded as an outer iteration for finding the root of the equation $\mu(\lambda) = \frac{1}{\lambda}$. Each iteration of the zero-finding requires a determination of $\mu_1(\lambda)$ (the largest generalized eigenvalue at a specific value of the parameter $\lambda$) such that $\|f\| e_1^T y_1 < \epsilon \mu_1$ ($y_1$ the eigenvector corresponding to $\mu_1(\lambda)$) through the MIRL method. Once the value of $\lambda$ is updated the next evaluation of $\mu_1$ will use the first column of $V$ to begin the new restarted Lanczos process. The latest eigenvalue, $\lambda$ from the zero-finder with the largest generalized eigenvalue from the eigen-solver will be used to initiate the next zero-finding step.

This chapter is concerned with zero-finders for one variable nonlinear functions. In the first two sections, we present some pre-existing root-finding techniques, such as Bisection, Newton-Raphson, Secant and Muller's method. These zero-finders were used for solving the nonlinear eigenvalue problems in [55], [40], [47] and [44], respectively. In the last section of this chapter we introduce a new zero-finder that is based on rational function interpolation to approximate the generalized eigenvalue $\mu(\lambda)$.

5.1 Some Techniques Involving the Use of a Derivative

Methods which make use of one or more of the derivatives behave rather differently in general from interpolation methods. The relative error in the computed derivative is much lower than that in the function value. This is generally true, even for multiple or very close zeros. But, the function is often only defined implicitly. Thus, we need some zero-finders that based on interpolation, such as the Muller's approach
or the secant method. Of course the bisection method neither involves derivatives nor depends on interpolation. For the sake of completeness, we propose the bisection technique in this section. We are going to use the notations \( \lambda_b, \lambda_m, \lambda_n \) and \( \lambda_s \) which stand for the solutions of Bisection, Muller, Newton-Raphson and secant techniques, respectively.

### 5.1.1 Bisection Method

Bisection method can be used in a general context to find real zeros of a real function. This method is based on systematically reducing the interval that contains the solution by function comparison. Suppose that an interval \([\lambda_1, \lambda_2]\) has been specified in which \(\mu(\lambda_1)\mu(\lambda_2) < 0\). We evaluate \(\mu\) at the midpoint of the interval and test its sign. If the function value is zero, the method terminates; otherwise, a new interval contains the solution is produced by discarding the value of \(\lambda_1\) or \(\lambda_2\), depending on whether \(\mu(\lambda_1)\) or \(\mu(\lambda_2)\) agrees in sign with \(\mu\) at the midpoint.

Bisection is guaranteed to find the solution within any specified tolerance \(\varepsilon\) if \(\mu\) can be computed to sufficient accuracy so that its sign is correct. The uncertainty interval may always be found by bisection under these conditions in about \(\log_2\left(\frac{\lambda_2-\lambda_1}{\varepsilon}\right)\) evaluations of \(\mu\). Bisection can be shown to be "optimal" method for the class of functions that change sign in \([\lambda_1, \lambda_2]\), in the sense that it yields the smallest interval that contains the solution for a specified number of function evaluations [5]. The following algorithm gives us one iteration from the Bisection method. To get the solution, we have to do this again until our criterion is satisfied.

**Algorithm 5.1** \(\text{function } [\lambda_b] = \text{Bisection}(\lambda_1, \lambda_2)\)

1. Set: \(\lambda_b \leftarrow \frac{\lambda_2+\lambda_1}{2}\);

2. If, \(\mu(\lambda_1)\mu(\lambda_2) \leq 0\) then \(\lambda_2 \leftarrow \lambda_b\);
   
   otherwise \(\lambda_1 \leftarrow \lambda_b\).
Some indication of the speed of the zero-finding algorithm can be obtained by finding the rate at which the interval that contains the solution converges to zero. For the Bisection algorithm, the length of this interval converges to zero linearly, with asymptotic error constant $\frac{1}{2}$.

5.1.2 Newton–Raphson’s Method

The defect of the Bisection method is that it takes no account of the relative magnitudes of the values of $\mu$ at the various points. If $\mu$ is known to be well behaved, it seems reasonable to use the values of $\mu$ at the endpoints to determine the next estimate of the solution. A mean of utilizing the magnitude of $\mu$ during the search for a zero is to approximate or model $\mu$. The zero of the model function $\hat{\mu}$ is taken to be a new estimate of the zero of the function $\mu$ itself. If $\mu$ is differentiable, an obvious candidate for $\hat{\mu}$ is the tangent line at the current estimate of the solution. The following algorithm describes one iteration of a technique which computes the zero of the tangent line.

Algorithm 5.2 function $[\lambda_n] = \text{NewtonR} (\lambda_1, \mu(\lambda_1), \mu'(\lambda_1))$

1. Set: $\Delta \leftarrow -\frac{\mu(\lambda_1)}{\mu'(\lambda_1)}$;
2. $\lambda_n \leftarrow \lambda_1 + \Delta$;
3. If $|\mu(\lambda_n)| \leq \epsilon$ then stop
   otherwise $\lambda_1 \leftarrow \lambda_n$ go to 1.

This iterative procedure is known as the Newton-Raphson Method for finding zero. The underlying assumption of this method is that, for the purpose of zero-finding, $\mu$ is "like" a straight line. One therefore expects Newton–Raphson’s method to work well when $\mu$ satisfies this assumption. In particular, if $\mu(\lambda_n)$ is nonzero and the starting point $\lambda_1$ is "close enough" to $\lambda_n$ this method converges quadratically to $\lambda_n$. The difficulty with this approach is that its convergence rate is only local and it depends
on the derivative of the function. Moreover, if $\lambda_1$ is not close enough to the solution, the Newton-Raphson iteration may not converge [30].

5.2 Some Techniques Based on Polynomial Interpolation

In the rest of this chapter, we concern ourselves with zero-finders that are based on interpolation of a function of one variable. Given a function $\mu(\lambda)$, one chooses a function $\Upsilon(\lambda)$ from among a certain class of functions (frequently, but not always, the class of polynomials of degree $n$ or less, for some $n$) such $\Upsilon(\lambda)$ agrees with $\mu(\lambda)$ at certain values of $\lambda$. These values of $\lambda$ are often referred to as “interpolation points”. One may specify that certain derivatives of $\Upsilon(\lambda)$ agree with the corresponding derivatives of $\mu(\lambda)$ at some of interpolation points. Interpolation is frequently used for the purpose of approximating the given function $\mu(\lambda)$ at the interpolation points. The zero-finders to be discussed in these sections are very useful for determining zeros for a real function, $\mu(\lambda)$. Compared to the Newton-Raphson method they have the advantage that the derivatives of $\mu$ need not be computed [50].

5.2.1 Secant Zero-finder

The secant technique is based on the linear interpolation. It may be described as follows: the mass and stiffness matrices are generated, the problem is formed in the standard form and the value of $\lambda$ is set to $\lambda = \lambda_1$. The linear eigenproblem

$$A(\lambda_1)x = \mu(\lambda_1)x \quad (5.1)$$

is solved. The next value of $\lambda$ is set to $\lambda_2 = \frac{1}{\mu_1(\lambda_1)}$ where, $\mu_1(\lambda_1)$ is the largest generalized eigenvalue of solving problem (5.1). The matrix $A(\lambda_2)$ is evaluated and a new linear eigensystem is solved to compute $\mu_1(\lambda_2)$. A straight line between $(\lambda_1, \mu_1(\lambda_1))$ and $(\lambda_2, \mu_1(\lambda_2))$ can be constructed. It intersects the curve $\frac{1}{\lambda}$ at the solution point. The point $(\lambda_1, \mu_1(\lambda_1))$ is set to be $(\lambda_2, \mu_1(\lambda_2))$ and the solution point is set to be $\lambda_2$. Again the matrix $A(\lambda_2)$ is computed and a linear eigenproblem is solved to compute
a new $\mu_1(\lambda_2)$ and construct another linear interpolation between $(\lambda_1, \mu_1(\lambda_1))$ and
$(\lambda_2, \mu_1(\lambda_2))$. We continue to do these secant approximations using the most updated
value of $\lambda$. For more details see [47]. The following algorithm describes one iteration
from the secant scheme.

**Algorithm 5.3**

function $[\lambda_s] = \text{Secant} (\lambda_1, \lambda_2, \mu_1(\lambda_1), \mu_1(\lambda_2))$

1. $\Phi = \frac{\mu_1(\lambda_2) - \mu_1(\lambda_1)}{(\lambda_2 - \lambda_1)}$;
2. $\Delta = \lambda_2 \Phi - \mu_1(\lambda_2)$;
3. $\lambda_1 \leftarrow \lambda_2; \mu_1(\lambda_1) \leftarrow \mu_1(\lambda_2)$;
4. $\delta = |\Delta^2 + 4\Phi|$;
5. $\lambda_s = \frac{(\Delta + \sqrt{\delta})}{2\Phi}$.  

This technique has convergence rate $\frac{1}{2}(1 + \sqrt{5})$, but it may take several function
evaluations to find a root. In addition, it may not match a highly nonlinear function
well because it is based on the linear interpolation.

### 5.2.2 Muller's Method

The methods discussed up to this point allow us to find an isolated zero of a function
once an approximation to that zero is known. These methods are not very satisfac-
tory when all the zeros of a function are required or when good initial approximations
are not available. For polynomial functions, there are methods which yield an ap-
proximation to all zeros simultaneously. Muller's method has been used to find any
prescribed number of zeros, real or complex of an arbitrary function. We are con-
cerned here with real zeros. Muller's method is an extension of the Secant method.
It is based on approximating the function in the neighborhood of the solution by a
quadratic polynomial. This gives a much closer match to the actual curve of the
function. A second-degree polynomial, $P_2(\lambda)$, is made to fit three points, $(\lambda_1, \mu(\lambda_1))$,  

(\lambda_2, \mu(\lambda_2)), (\lambda_3, \mu(\lambda_3)), near a root and the proper zero of this quadratic, using the quadratic formula, is used as the improved estimate of the root being evaluated. The procedure of Muller's method is developed by using a quadratic equation that fits through three points in the vicinity of a zero, in the form \(a\lambda^2 + b\lambda + c = 0\). The development is simplified if we transform axes to pass through the middle point, by letting \(\xi = \lambda - \lambda_1\). Evaluating \(P_2(\xi)\) at the three points. After computing \(\alpha, \beta\) and \(\gamma\), we solve for the root of \(\alpha\xi^2 + \beta\xi + \gamma = 0\) by the quadratic formula, choosing the zero nearest to the middle point \(\lambda_2\). We take the root of the polynomial as one of a set of three points for the next approximation, taking the three points that are most closely spaced (that is, if the zero is to the right of \(\lambda_1\), take \(\lambda_1, \lambda_2\) and the root; if to the left, take \(\lambda_1, \lambda_3\) and the root). We always reset the subscript to make \(\lambda_1\) be the middle point of the three values. The following algorithm gives a complete iteration from this technique.

**Algorithm 5.4**  

\[
\text{function } [\lambda_m] = \text{Muller} (\lambda_1, \lambda_2, \lambda_3) 
\]

1. **Evaluate** the corresponding function values \(\mu(\lambda_1), \mu(\lambda_2)\) and \(\mu(\lambda_3)\);
2. **Find** the coefficients of the parabola, \(a\lambda^2 + b\lambda + c = 0\) determined by the three points;
3. **Compute** the two roots of the quadratic equation, \(\alpha\xi^2 + \beta\xi + \gamma = 0\) where, \(\xi = \lambda - \lambda_1\);
4. **Choose** the root closest to \(\lambda_1\) and label it \(\lambda_m\);
5. If, \((\lambda_m > \lambda_1), \lambda_1 \leftarrow \lambda_3, \lambda_m \leftarrow \lambda_1\)  
   otherwise, \(\lambda_1 \leftarrow \lambda_2, \lambda_m \leftarrow \lambda_1\);
6. If \(|\mu(\lambda_m)| < \epsilon\), then stop  
   else, go to 1.

This method converges almost quadratically in the vicinity of a root. It also does not require the evaluation of the function's derivative. In addition, it obtains both
real and complex roots even when these roots are not simple. Unfortunately, this algorithm breaks down if the choice of the interval is not optimal [8].

In fact, we have tried many zero-finding techniques other than those discussed here. These include:

- **HYBRID** from MINPACK which is used for solving a system of nonlinear equations and is reduced to the secant method in one dimension. This routine is based on the trust region globalization strategy and uses finite differences for approximating the Jacobian matrix [34].

- **ZEROIN** from TOMS is a routine used for minimizing a nonlinear function without derivatives. It is a combination of the bisection method and the linear and the inverse interpolations [2].

We have found that these techniques to be inadequate for our needs.

### 5.3 A new Technique Based on Rational Interpolation

In spite of the fact that any function $\mu(\lambda)$ which is continuous on a closed interval can be approximated to an arbitrary degree of accuracy on the interval by some polynomial, the use of polynomial approximation may not be satisfactory for many functions. The degree of the polynomial required to attain a given accuracy of approximation may be excessively high [58]. In some cases, the situation is different if the location $\lambda$ for which one desires an approximation value of $\mu(\lambda)$ lies in the proximity of a pole or some other singularity of $\mu(\lambda)$ [3]. For example, consider the value $\tan(\lambda)$ for $\lambda$ close to $\frac{\pi}{2}$. One might expect that in such a case rational function interpolation would give a more accurate approximation than a polynomial of corresponding degree because rational functions themselves may have poles. As a matter of fact, it has been found that in many cases considerably greater accuracy can be obtained for a given amount of computational effort by the use of rational functions than by the use of polynomials. This is borne out by an inspection of the tables of polynomial and rational function approximations given by Hart, et al. (1968) for a
variety of mathematical functions [53]

**Rational Function Zero-finder**

The structure of the matrix $A(\lambda)$ arising from the use of the frequency dependent shape functions in the finite element method includes matrix entries that are composed of elementary transcendental functions approximated well by rational functions. These features are reflected in the functions $\mu_j(\lambda)$ so that special techniques must be derived to ensure convergence to physically meaningful results. The well-known secant method discussed in the previous subsection gives remarkable results to the solution of equation $\mu(\lambda) = \frac{1}{\lambda}$. However, it still takes several iterations to find a root. Methods based upon the rational function interpolation have been used to great advantage in solving trust region problems [24]. The value $\mu_j(\lambda)$ is a root of a polynomial whose coefficients are rational functions of the matrix elements $A(\lambda)$. The solution of the dependent problem is the intersection point between the two curves $\mu(\lambda)$ and $\frac{1}{\lambda}$ (see figures 3.1 and 3.2). The approximation of the function $\mu(\lambda)$ can be written as:

$$\mu(\lambda) \approx \frac{a\lambda + \beta}{\gamma\lambda + 1} \quad (5.2)$$

which has the advantage of being linear when $\gamma$ equals to zero, [12]. The constants $a$, $\beta$, and $\gamma$ are determined by interpolating $\mu(\lambda)$ at three distinct points. The following is a description for constructing these three points.

1. **Initialize** the eigenvalue $\lambda$ with $\lambda_1$;

2. **Solve** : $A(\lambda_1)x = \mu(\lambda_1)x$; $\mu_1 \leftarrow \max \mu(\lambda_1)$; $\lambda_2 \leftarrow \frac{1}{\mu_1}$;

3. **Solve** : $A(\lambda_2)x = \mu(\lambda_2)x$; $\mu_2 \leftarrow \max \mu(\lambda_2)$;

4. **Apply** : secant for one step between $(\lambda_1, \mu_1)$ and $(\lambda_2, \mu_2)$ to find secant point $\lambda_s$;

5. **Determine** : $\lambda_3 \leftarrow \lambda_2, \mu_3 \leftarrow \mu_2, \lambda_2 \leftarrow \lambda_3$ and $\mu_2 \leftarrow \mu_1 + \epsilon (\epsilon = 10^{-6})$;
The rational interpolation between the three points \((\lambda_1, \mu_1), (\lambda_2, \mu_2)\) and \((\lambda_3, \mu_3)\) is applied to compute the constants \(\alpha, \beta, \gamma\). The larger root of the quadratic \(\alpha \lambda^2 - (\gamma - \beta) \lambda - 1 = 0\) is the updated value of \(\lambda\). The next algorithm introduces a complete iteration of the rational function procedure.

**Algorithm 5.5**

function \([\lambda_1, \lambda_2, \lambda_3] = \text{Rational}(\lambda_1, \lambda_2, \lambda_3, \mu_1, \mu_2, \mu_3)\);

1. Solve
   \[
   \begin{pmatrix}
   1 & \lambda_1 & -\mu_1 \lambda_1 \\
   1 & \lambda_2 & -\mu_2 \lambda_2 \\
   1 & \lambda_3 & -\mu_3 \lambda_3
   \end{pmatrix}
   \begin{pmatrix}
   \alpha \\
   \beta \\
   \gamma
   \end{pmatrix}
   =
   \begin{pmatrix}
   \mu_1 \\
   \mu_2 \\
   \mu_3
   \end{pmatrix}
   \]
   for \(\alpha, \beta, \gamma\);

2. Set \(\lambda_1 \leftarrow \lambda_2, \mu_1 \leftarrow \mu_2\);

3. Solve: \(\alpha \lambda^2 - (\gamma - \beta) \lambda - 1 = 0\);
   \[
   \lambda_2 \leftarrow \frac{(\gamma - \beta) + \sqrt{(\gamma - \beta)^2 + 4 \alpha}}{2 \alpha}.
   \]

A new iteration starts by solving a new linear eigenvalue problem at \(\lambda_2\) (the most updated value resulting from the previous iteration). Apply Secant between \((\lambda_2, \mu_2)\) and \((\lambda_3, \mu_3)\) to find the Secant point, \(\lambda_s\), then apply Rational zero-finder between the new points \((\lambda_2, \mu_2), (\lambda_3, \mu_3)\) and \((\lambda_s, \mu_1 + \epsilon), \epsilon = 10^{-6}\) and so on.

This strategy has some drawbacks that can lead to non-convergence. These problems are described as follows:

1. The matrix in the linear system in algorithm 5.5 becomes
   \[
   \begin{pmatrix}
   1 & \lambda_1 & -\mu_1 \lambda_1 \\
   0 & \lambda_2 - \lambda_1 & \mu_1 \lambda_1 - \mu_2 \lambda_2 \\
   0 & 0 & \theta
   \end{pmatrix}
   \]
   where, \(\theta = (\mu_1 \lambda_1 - \mu_3 \lambda_3) - (\mu_1 \lambda_1 - \mu_2 \lambda_2) (\frac{\lambda_2 - \lambda_1}{\lambda_3 - \lambda_1})\) after applying Gauss elimination. This matrix may lead to an ill-conditioned linear system when one of
the diagonal entries blows up very fast or it becomes very close to zero. This situation occurs when the values of the function \( \mu(\lambda) \) are getting very small or the difference between \( \lambda_2 \) and \( \lambda_1 \) is very close to zero.

2. The scale dependent way in which \( \mu_2 \) has been chosen is not efficient. We need a scale independent way that depends on the physical properties of the system.

3. The rational approximation \( \mu(\lambda) \approx \frac{\alpha + \beta}{\gamma + 1} \) has a pole \( -\frac{1}{\gamma} \) which is one of the advantages of rational function approximation. The pole property of the rational function may lead to a problem when the approximated function has no pole. In this situation, we want to reduce the function to a quadratic behavior.
Chapter 6

THE SAFEGUARDED TECHNIQUE

The best methods available for zero-finding are the so-called safeguarded procedures. The idea is to combine a guaranteed, reliable method (such as bisection or linear interpolation) with a rapidly convergent method (such as rational interpolation), to yield an algorithm that will converge at least with the rate of the safe guaranteed method if \( p \) is well-behaved [50].

A safeguarded interpolation method might include the following logic at each iteration. Using enough known information to find a new point without safeguards, the next step would evaluate the function value at the new point, and discard one of the old points to form a new point. However, a safeguarded procedure ensures that the new point is a "reasonable" point before evaluating the function at that point. The point is reasonable when it lies in the uncertainty interval (i.e. it contains the solution) and it guarantees that the number of function evaluations required should not be significantly more than that required by the safeguard.

Finally, safeguards are necessary to ensure that successive iterates are not "too close". It can happen that the new point is numerically indistinguishable from the previous best point, even when the new point is far from optimal. Even if the new point were accepted, no further progress would occur [17].

This chapter consists of three parts. First, we treat the problems that appeared when we used the rational zero-finder in the previous chapter. Second, we modify the rational zero-finder to be safeguarded, scale independent and to control the pole. Third, we develop our new safeguarded technique for solving the nonlinear eigenproblems associated with the mixed finite element model.
6.1 The Simplified Preferable Approach

In order to overcome the difficulties exposed in the previous chapter, we simplify the development by transforming the axes to pass through the point "\(\lambda_2\)”, where \(\lambda_2 \in (\lambda_1, \lambda_3)\). The function \(\mu(\lambda) \approx \frac{\alpha\lambda + \beta}{\gamma + 1}\) can be interpolated at the points \((\lambda_1, \mu_1), (\lambda_2, \mu_2)\) and \((\lambda_3, \mu_3)\) to evaluate the coefficients \(\alpha, \beta\) and \(\gamma\). These values can be evaluated as follows:

\[
\beta = \mu_2 \\
\alpha = \frac{1}{\delta_1 + \delta_2} \left[ \frac{(\delta_1 + \delta_2)\mu_2 - (\delta_2\mu_1 + \delta_1\mu_3)}{\delta_1\delta_2(\mu_3 - \mu_1)} \right] (\delta_1\mu_1 + \delta_2\mu_3) + (\mu_3 - \mu_1) \\
\gamma = \frac{(\delta_1 + \delta_2)\mu_2 - (\delta_2\mu_1 + \delta_1\mu_3)}{\delta_1\delta_2(\mu_3 - \mu_1)}
\]

where, \(\delta_1 = \lambda_2 - \lambda_1, \delta_2 = \lambda_3 - \lambda_2\) and \(\mu_i = \mu(\lambda_i), i = 1, 2, 3\).

First, we satisfy the condition \((\delta_1\mu_1 - 1)(\delta_2\mu_3 - 1) \leq 0\) to ensure that \(\delta_2\) is large enough. We also guarantee that the interval \([\delta_1, \delta_2]\) is the uncertainty interval. In the neighborhood of the solution, \(\delta_2\) will be very small. This would cause the value of \(\gamma\) to blow up. In this situation, we either use the safeguard or shrink the step. The value \(\mu_3 - \mu_1\) may lead to the same problem. In this case we employ the safeguard.

The choice of \(\mu_2\) needs to be scale independent. We may interpolate linearly or use the Bisection method (for one step only) between the two points \((\lambda_1, \mu_1)\) and \((\lambda_3, \mu_3)\) to evaluate \(\lambda_2\). The solution of the linear eigenvalue problem

\[
A(\lambda_2)x = \mu(\lambda_2)x \quad \text{gives} \quad \mu_2 = \max \mu(\lambda_2).
\]

This is expensive because it increases the number of function evaluations. The cheapest way to compute \(\mu_2\) is to use the information in hand so that we do not need to solve a new linear eigenvalue problem. The line \(\lambda = \lambda_2\) intersects the curve \(\frac{1}{\lambda}\) in the point \((\lambda_2, \frac{1}{\lambda_2})\) which gives the upper bound, \(\frac{1}{\lambda_2}\) for the value \(\mu_2\). The function \(\mu(\lambda)\) is monotone increasing on the uncertainty interval. Its values are the generalized eigenvalues of symmetric positive definite matrices. Thus, the value \(\mu_2\) has a lower
bound \( \mu_1 \) on the interval \([\lambda_1, \lambda_3]\), see figure 6.2. The choice of \( \mu_2 \) must lie in the interval \([\mu_1, \frac{1}{\lambda_2}]\) and that is why the previous rational technique works (the choice of the third point was, \( \mu_2 = \mu_1 + \epsilon \)). The function \( \frac{\alpha \lambda + \beta}{\gamma \lambda + 1} \) has a pole at \( \frac{-1}{\gamma} \). So, for some choices of \( \mu_2 \in [\mu_1, \frac{1}{\lambda_2}] \) the pole may lie inside the uncertainty interval, \([\lambda_1, \lambda_3]\). The best choice of \( \mu_2 \) is the value that lies in \([\mu_1, \frac{1}{\lambda_2}]\) and guarantees that the pole lies outside the uncertainty interval.

Any value of the function \( \mu(\lambda) \) is a generalized eigenvalue of the symmetric positive definite matrix, \( A(\lambda) \) at a specific value of the parameter \( \lambda \). Since, \( \mu(\lambda) \approx \frac{\alpha \lambda + \beta}{\gamma \lambda + 1} \), where \( \delta = \lambda - \lambda_2 \) is always positive quantity then

\[
\alpha \delta + \beta > 0 \quad \text{and} \quad \gamma \delta + 1 > 0 \tag{6.4}
\]

or

\[
\alpha \delta + \beta < 0 \quad \text{and} \quad \gamma \delta + 1 < 0 \tag{6.5}
\]

From inequalities (6.4) we have \( \delta > \frac{-1}{\gamma} \). We also know that \( \delta \) is always positive so that \( \gamma > 0 \) or \( \gamma < 0 \). From inequalities (6.5), we get \( \delta < \frac{-1}{\gamma} \) which is true when \( \gamma < 0 \). From this discussion we have the following important remarks.

1. The pole lies away from the right end of the interval of uncertainty, \([\delta_1, \delta_2]\), when \( \delta_2 < \frac{-1}{\gamma} \) and \( \gamma < 0 \).

2. If \( -\delta_1 > \frac{-1}{\gamma} \) and \( \gamma > 0 \) then the pole lies away from the left end of the uncertainty interval, \([\delta_1, \delta_2]\).

The following lemmas determine conditions on the third point, \( \mu_2 \). These conditions guarantee that the pole lies outside the uncertainty interval.

**Lemma 6.1** The pole of the rational function \( \mu(\lambda) \approx \frac{\alpha \delta + \beta}{\gamma \delta + 1} \) lies away from the right end of the interval \([\delta_1, \delta_2]\) if and only if

\[
\mu_2 > \mu_1 \quad \text{and} \quad \gamma < 0.
\]
Proof: Suppose that the pole lies away from the right end of the interval $[\delta_1, \delta_2]$ " $\delta_2 < \frac{1}{\gamma}$ and $\gamma < 0$" then, $\gamma \delta_2 + 1 > 0$. But $\mu_3 \approx \frac{\delta_2 + \beta}{\gamma \delta_2 + 1}$ is always positive. In other words, $\mu_3$ is a generalized eigenvalue for a positive definite matrix, so it may be approximated by a positive quantity then,

$$\alpha \delta_2 + \beta > 0. \quad (6.6)$$

Introducing equations (6.1) and (6.2) in the inequality (6.6) we get

$$\delta_2 \left[ \frac{\mu_2 (\delta_1 \mu_1 + \delta_2 \mu_3)}{\delta_1 \delta_2 (\mu_3 - \mu_1)} - \frac{(\delta_2 \mu_1 + \delta_1 \mu_3)(\delta_1 \mu_1 + \delta_2 \mu_3)}{(\delta_1 + \delta_2) \delta_1 \delta_2 (\mu_3 - \mu_1)} + \frac{\mu_3 - \mu_1}{\delta_1 + \delta_2} \right] > -\mu_2. $$

Using the previous inequality and rearranging terms give

$$\mu_2 \left[ \frac{\delta_2 (\delta_1 \mu_1 + \delta_2 \mu_3)}{\delta_1 \delta_2 (\mu_3 - \mu_1)} + 1 \right] > \frac{\delta_2 (\delta_2 \mu_1 + \delta_1 \mu_3)(\delta_1 \mu_1 + \delta_2 \mu_3)}{(\delta_1 + \delta_2) \delta_1 \delta_2 (\mu_3 - \mu_1)} - \frac{\delta_2 (\mu_3 - \mu_1)}{\delta_1 + \delta_2}. $$

After rearranging the terms in the last inequality we get

$$\mu_2 [\delta_2 (\delta_1 \mu_1 + \delta_2 \mu_3) + \delta_1 \delta_2 (\mu_3 - \mu_1)] > \frac{\delta_2}{\delta_1 + \delta_2} [(\delta_2 \mu_1 + \delta_1 \mu_3)(\delta_1 \mu_1 + \delta_2 \mu_3) - \delta_1 \delta_2 (\mu_3 - \mu_1)^2]. $$

After eliminating the similar terms in the right hand side of the above inequality, we get

$$\mu_2 [(\delta_1 \mu_1 + \delta_2 \mu_3) + \delta_1 (\mu_3 - \mu_1)] > (\delta_1 + \delta_2) \mu_1 \mu_3. $$

From this inequality and the assumption $\gamma < 0$, we have

$$\mu_2 > \mu_1 \quad \text{and} \quad \gamma < 0 \quad (6.7)$$

For proving the other direction of the lemma, suppose that condition (6.7) is satisfied.

Using (6.1) and (6.2) and rearranging terms give

$$\alpha \delta_2 + \beta = \mu_2 \left[ \frac{\delta_2 (\delta_1 \mu_1 + \delta_2 \mu_3) + \delta_1 \delta_2 (\mu_3 - \mu_1)}{\delta_1 \delta_2 (\mu_3 - \mu_1)} \right] - \delta_2 \left[ \frac{(\delta_2 \mu_1 + \delta_1 \mu_3)(\delta_1 \mu_1 + \delta_2 \mu_3)}{(\delta_1 + \delta_2) \delta_1 \delta_2 (\mu_3 - \mu_1)} + \frac{\mu_3 - \mu_1}{\delta_1 + \delta_2} \right].$$

Using inequality, $\mu_2 > \mu_1$ and rearranging terms give

$$\alpha \delta_2 + \beta > \frac{\delta_2 (\delta_1 + \delta_2) \mu_1 \mu_3}{\delta_1 \delta_2 (\mu_3 - \mu_1)} - \frac{(\delta_2 \mu_1 + \delta_1 \mu_3)(\delta_1 \mu_1 + \delta_2 \mu_3)}{(\delta_1 + \delta_2) \delta_1 \delta_2 (\mu_3 - \mu_1)} + \frac{\mu_3 - \mu_1}{\delta_1 + \delta_2}. $$
The right hand side of this inequality is zero. So, \( \alpha \delta_2 + \beta > 0 \) but \( \mu(\delta_2) > 0 \) then \( \gamma \delta_2 > -1 \). Since \( \delta_2 \) is positive, hence \( \gamma > 0 \) or \( \gamma < 0 \). The inequalities \( \delta_2 < -\frac{1}{\gamma} \) and \( \gamma < 0 \) assure that the pole lies away from the right end of the uncertainty interval \([\delta_1, \delta_2]\). The other possibility \((\delta_2 > -\frac{1}{\gamma} \text{ and } \gamma > 0)\) is rejected because the pole may lie inside the interval \([\delta_1, \delta_2]\).

**Lemma 6.2** The pole of the rational function \( \mu(\lambda) \approx \frac{\alpha \lambda + \beta}{\gamma \delta + 1} \) lies to the left end of the interval \([\delta_1, \delta_2]\) if and only if

\[
\mu_2 > \mu_3 \quad \text{and} \quad \gamma > 0
\]

**Proof:** Suppose that the pole lies away from the left end of the interval \([\delta_1, \delta_2]\). So, \( -\delta_1 > -\frac{1}{\gamma} \) and \( \gamma > 0 \). Since \( \mu_1 \approx \frac{\alpha \delta_1 + \beta}{\gamma \delta_1 + 1} \) is positive and \( -\gamma \delta_1 + 1 > 0 \), then \( -\alpha \delta_1 + \beta > 0 \). Using inequalities (6.1) and (6.2) and rearranging the terms give

\[
-\alpha \delta_1 + \beta = -\delta_1 \left[ \frac{\mu_2(\delta_1 \mu_1 + \delta_2 \mu_3)}{\delta_1 \delta_2(\mu_3 - \mu_2)} - \frac{(\delta_2 \mu_1 + \delta_1 \mu_3)(\delta_1 \mu_1 + \delta_2 \mu_3)}{(\delta_1 + \delta_2) \delta_1 \delta_2(\mu_3 - \mu_1)} + \frac{\mu_3 - \mu_1}{\delta_1 + \delta_2} \right] + \mu_2
\]

But, \( -\alpha \delta_1 + \beta > 0 \) then

\[
\mu_2 [-\delta_1(\delta_1 \mu_1 + \delta_2 \mu_3) + \delta_1 \delta_2(\mu_3 - \mu_1)] > -\frac{\delta_1}{\delta_1 + \delta_2} \left[ (\delta_2 \mu_1 + \delta_1 \mu_3)(\delta_1 \mu_1 + \delta_2 \mu_3) - \delta_1 \delta_2(\mu_3 - \mu_1)^2 \right]
\]

From this inequality we get

\[
\mu_2 > \frac{-\delta_1(\delta_1 + \delta_2) \mu_1 \mu_3}{-\delta_1(\delta_1 \mu_1 + \delta_2 \mu_3) + \delta_1 \delta_2(\mu_3 - \mu_1)}, \quad \text{i.e.} \quad \mu_2 > \mu_3 \quad \text{and} \quad \gamma > 0 \quad (6.8)
\]

The proof of the second direction of the lemma can be derived by assuming the condition (6.8). Using formulas (6.1) and (6.2) and rearranging terms give

\[
-\alpha \delta_1 + \beta \geq -\delta_1 \left( \frac{\delta_1 + \delta_2}{\delta_1 \delta_2(\mu_3 - \mu_1)} + \delta_1 \frac{(\delta_2 \mu_1 + \delta_1 \mu_3)(\delta_1 \mu_1 + \delta_2 \mu_3)}{(\delta_1 + \delta_2) \delta_1 \delta_2(\mu_3 - \mu_1)} - \frac{\mu_3 - \mu_1}{\delta_1 + \delta_2} \right)
\]

The right hand side of this inequality equals zero, then \( -\alpha \delta_1 + \beta > 0 \) but \( \mu_1 \) is positive. So, we deduce that \( -\delta_1 > -\frac{1}{\gamma} \) with \( \gamma > 0 \) which completes the proof. ■

The behavior of the function \( \mu(\lambda) \) is monotone increasing (figure 4.3) and the shape of
the rational approximation \( \frac{\alpha \lambda^2 + \beta}{\gamma \lambda + 1} \) in figure 6.1 lead us to choose the left branch of this rational approximation to approximate our function. So, we are sure that the pole of this approximation should be away from the right end of the uncertainty interval \([\delta_1, \delta_2]\).

### 6.2 Safeguarded Rational Zero-finding Technique

The safeguarded rational zero-finder is the modification of the previous one (algorithm 5.5). This technique is based upon rational function interpolation. It has many advantages:

1. The evaluation of \( \mu_2 \) in a scale independent way guarantees that the pole lies outside the interval of uncertainty. Moreover, we do not need to solve more linear eigenproblems.

2. Secant method is used as a safeguard when the values of the function \( \mu(\lambda) \) is getting small, i.e, the linear system is ill-conditioned.

3. This technique has the flexibility to search between \( \lambda_2 \) and \( \lambda_3 \) as well as between \( \lambda_1 \) and \( \lambda_2 \). In addition, it gives a much closer match to the actual curve of the function \( \mu(\lambda) \).

The three points that are required for the first iteration in this procedure may be described as follows:

1. **Initialize**: \( \lambda \) to \( \lambda_1 \) which may be zero;

2. **Solve**: \( A(\lambda_1)x = \mu(\lambda_1)x; \mu_1 \leftarrow \max \mu(\lambda_1); \lambda_2 \leftarrow \frac{1}{\mu_1}; \)

3. **Solve**: \( A(\lambda_2)x = \mu(\lambda_2)x; \mu_2 \leftarrow \max \mu(\lambda_2) \).
4. Evaluate: a new value for \( \lambda \) that lies in the interval \((\lambda_1, \lambda_2)\)

\[
\phi = \frac{\mu_2 - \mu_1}{\lambda_2 - \lambda_1}
\]

\[
\Delta = \lambda_2 \phi - \mu_2
\]

\[
\lambda_3 \leftarrow \lambda_2, \text{ and } \mu_3 \leftarrow \mu_2
\]

\[
\eta = |\Delta^2 + 4\phi|
\]

\[
\lambda_2 = \frac{(\Delta + \sqrt{\eta})}{2\phi}
\]

5. Determine: the value of the third point in a scale independent way,

\[
\mu_2 = 0.5 \left( \frac{1}{\lambda_2} + \mu_1 \right).
\]

Once the values of the three successive points \((\lambda_1, \mu_1), (\lambda_2, \mu_2)\) and \((\lambda_3, \mu_3)\) are found, the rational approximation \(\mu(\lambda) \approx \frac{\beta + \alpha \lambda}{\gamma + \lambda} \) is applied for evaluating the values of the coefficients \(\alpha, \beta,\) and \(\gamma\). The larger root of the quadratic \(\alpha \delta^2 - (\gamma - \beta) \delta - 1 = 0\) is computed. The following algorithm is a description for a complete iteration from the safeguarded procedure:

Algorithm 6.1 function \([\lambda_1, \lambda_2, \lambda_3] = \text{safational} (\lambda_1, \lambda_2, \lambda_3, \mu_1, \mu_2, \mu_3);\)

(1) Check Singularity:

If \((\mu_2 \leq \epsilon_1)\) then,

\[
\lambda_2 \leftarrow \text{Secant } [\lambda_1, \lambda_2, \mu_1, \mu_2]
\]

return.

(2) Determine the third point:

\[
\phi = \frac{\mu_2 - \mu_1}{\lambda_2 - \lambda_1}
\]

\[
\Delta = \lambda_2 \phi - \mu_2
\]

\[
\lambda_3 \leftarrow \lambda_2, \text{ and } \mu_3 \leftarrow \mu_2
\]

\[
\eta = |\Delta^2 + 4\phi|
\]

\[
\lambda_2 = \frac{(\Delta + \sqrt{\eta})}{2\phi}
\]

\[
\delta_1 = \lambda_2 - \lambda_1, \delta_2 = \lambda_3 - \lambda_2
\]

\[
\mu_2 = 0.5 \left( \frac{1}{\lambda_2} + \mu_1 \right).
\]
(3) Solve :
\[
\begin{pmatrix}
-\delta_1 & 1 & \delta_1 \mu_1 \\
0 & 1 & 0 \\
\delta_2 & 1 & -\delta_2 \mu_3
\end{pmatrix}
\begin{pmatrix}
\alpha \\
\beta \\
\gamma
\end{pmatrix}
= 
\begin{pmatrix}
\mu_1 \\
\mu_2 \\
\mu_3
\end{pmatrix}
\]

For \(\alpha, \beta\) and \(\gamma\).

(4) Set : \(\xi = \delta_1 \delta_2 (\mu_3 - \mu_1)\)
\[
\lambda_1 \leftarrow \lambda_2; \mu_1 \leftarrow \mu_2
\]

(5) Solve :
\[
\alpha \delta^2 - (\gamma - \beta) \delta - 1 = 0;
\]
\[
\delta \leftarrow \frac{(\gamma - \beta) + \sqrt{(\gamma - \beta)^2 + 4 \alpha}}{2 \alpha};
\]

(6) If \((\delta > \delta_1 \text{ and } \xi \leq \epsilon_2)\) then
\[
\lambda_2 = \lambda_1 + \epsilon_2 \delta_2
\]
else if \((\delta > \delta_1 \text{ and } \xi > \epsilon_2)\) then
\[
[\lambda_2] \leftarrow \text{Secant}(\lambda_2; \lambda_3; \mu_2; \mu_3)
\]
else
\[
\lambda_2 = \lambda_1 + \delta
\]
end if.

A new iteration starts, the matrix \(A(\lambda)\) is computed at the most updated value of \(\lambda\) which is \(\lambda_2\) (that comes from the zero-finder) and a linear eigenvalue problem is solved to evaluate the value \(\mu_2 = \max \mu(\lambda_2)\). A new point between the two points \((\lambda_2, \mu_2)\) and \((\lambda_3, \mu_3)\) can be evaluated using the secant method. Replace the point \((\lambda_2, \mu_2)\) by \((\lambda_1, \mu_1)\) and the secant point \(\lambda_s\) by \(\lambda_2\). The value \(\mu_2\) is computed and then the safeguarded rational zero-finder is called. This process is continued until the stopping criteria is satisfied and the first eigenvalue, \(\lambda_1^*\) of the nonlinear eigenproblem is computed. Once the solution to the \(j\)-th nonlinear problem \(\mu_j(\lambda) = \frac{1}{\lambda}\) under the condition \(\|f_k\| e_k^T y_j < \epsilon \mu_j\) (where \(\mu_j = \max \mu(\lambda_j)\), the largest generalized eigenvalue and its corresponding eigenvector, \(y_j\)) has been found. The \(k\)-step Lanczos factorization from the final iterative step of solving the \(j\)-th problem will be used to
initiate the \((j+1)\)-th problem. The solution to the \(j\)-th nonlinear problem, \(\lambda_j^*\) provides an excellent initial guess to begin the zero-finding iteration for the next desired eigenvalue, \(\lambda_{j+1}^*\). In this algorithm, we used \(\epsilon_1 = 5 \times 10^{-2}\) and \(\epsilon_2 = 10^{-5}\).

6.3 The New Approach

The desired nonlinear eigenvalue problem, (3.4) has a special structure and the goal of this technique is to compute a few of the smallest eigenvalues and their corresponding eigenvectors. The use of the formulated finite element model [23] saves the computations because the matrix \(K\) is factored only once and this factorization is stored for all iterations. The new approach is based on the well known idea linearize, solve and then update. The eigen-solver is the MIRL method, (3.7) that has been determined to be well suited for solving the linear eigenproblems. The MIRL has a close connection with the \(QR\) iteration and the ability to maintain an orthogonal basis of a predetermined size. The convergence of IRL is rapid for evaluating the largest few generalized eigenvalues, \(\mu_j(\lambda)\) for specific values of the parameter \(\lambda\). The safeguarded zero-finder is used to update the eigenvalue \(\lambda\). This zero-finder is based on rational interpolation. The convergence rate of these types of zero-finders is very close to the convergence rate of the Newton-Raphson method, as reported in [50]. Moreover, there is no need for the derivative of the function. The next algorithm introduces a complete iteration from this new scheme. The starting value \(\lambda_1\) may be chosen to be zero at the beginning or it is already computed from the previous iteration. This value \(\lambda_1\) is used to compute the mass matrix \(M(\lambda_1)\) and the MIRL method is applied to solve the linear eigenvalue problem

\[
A(\lambda_1)x = \mu_1(\lambda_1)x \quad \text{where,} \quad A(\lambda_1) = L^{-1}M(\lambda_1)L^{-T}.
\]

The largest generalized eigenvalue \(\mu_1(\lambda_1)\) for that linear problem is computed. The second point \((\lambda_2, \mu_2)\) is set to be \((\lambda_2, \mu_2) = (\frac{1}{\mu_1(\lambda_2)}, \mu_1(\lambda_2))\), where \(\mu_1(\lambda_2)\) is the largest generalized eigenvalue of solving a linear eigenproblem at \(\lambda_2\). The secant technique
is applied and $\mu_2$ can be computed. The safeguarded rational technique is applied between the points $(\lambda_1, \mu_1)$, $(\lambda_2, \mu_2)$ and $(\lambda_3, \mu_3)$ to evaluate the most updated value of $\lambda$. This process is continued until the stopping criteria satisfied. The following is a full description of a complete iteration of the new algorithm.

**Algorithm 6.2**

function $[\lambda_2, V] = \text{safeLanczos} (K, M(\lambda), k, p, tol, \lambda_1);$

1. **Factor**: $K$ into $LL^T$
2. **Compute**: $M(\lambda_1);$
3. $[T, V, f] \leftarrow \text{MIRL}(M(\lambda_1), L, k, p, tol);$
4. $\mu_1 \leftarrow v_{k+p}^T v_{k+p},$
   
   $\lambda_2 \leftarrow \frac{1}{\mu_1};$
5. **Compute**: $M(\lambda_2);$
6. $[T, V, f] \leftarrow \text{MIRL}(M(\lambda_2), L, k, p, tol);$
7. $\mu_2 \leftarrow v_{k+p}^T v_{k+p};$
8. **Check the uncertainty of the interval**
   
   If $((\lambda_1 \mu_1 - 1)(\lambda_2 \mu_2 - 1) > 0)$ then
   
   $\lambda_2 \leftarrow \frac{3}{2} \lambda_2$ go to 5;
9. **Update $\lambda$**

   $[\lambda_1, \lambda_2, \lambda_3] \leftarrow \text{saferational}(\lambda_1, \lambda_2, \lambda_3, \mu_1, \mu_2, \mu_3);$
10. **Check Convergence**:

    If, $|\lambda_2 - \lambda_1| \leq 0.5\epsilon(\lambda_1 + \lambda_2)$, then stop;
11. **Compute**: $M(\lambda_2);$
12. $[T, V, f] \leftarrow \text{MIRL}(M(\lambda_2), L, k, p, tol);$

     $\mu_2 \leftarrow v_{k+p}^T v_{k+p};$
\begin{equation}
\lambda_1 \leftarrow \lambda_2, \mu_1 \leftarrow \mu_2,
\lambda_2 \leftarrow \lambda_3, \mu_2 \leftarrow \mu_3;
\end{equation}
go to (9)

This process continues until the stopping criterion is satisfied and the first smallest eigenvalue, \(\lambda_1^*\) is computed. The second eigenvalue \(\lambda_2^*\) can be computed by initializing the eigen-solver and the zero-finder using the previous information. The initial vector \(v_1\) for the MIRL will be the first column of the matrix \(V\) from the last \(k\)-step factorization from the previous iteration. We solve the linear eigenvalue problem

\[ A(\lambda_2)x = \mu(\lambda_2)x \quad (6.9) \]

where, \(\lambda_2 = \frac{1}{\mu_2(\lambda_1)}\) and \(\mu_2(\lambda_1)\) is the second largest generalized eigenvalue of the last linear eigenvalue problem has been solved for finding \(\lambda_1^*\). The two points that the zero-finder needs are set to be \((\lambda_1, \mu_1) = (\lambda_1^*, \mu_2(\lambda_1))\) and \((\lambda_2, \mu_2) = (\frac{1}{\mu_2(\lambda_1)}, \mu_2(\lambda_2))\), where \(\mu_2(\lambda_2)\) is the second largest generalized eigenvalue of problem (6.9).

This strategy takes at least solution of two linear eigenvalue problems to compute the first nonlinear eigenvalue, \(\lambda_1^*\). It also takes at least solving one linear eigenvalue problem to find each one from the rest, \(\lambda_j^*, j = 2, 3, \ldots, 10\).
Figure 6.1 An example for a rational function

In the following figure, the intersection points of the secant line with $\mu(\lambda)$ and $\frac{1}{\lambda}$ curves are: $(\lambda_1, \mu_1), (\lambda_2, \frac{1}{\lambda_2})$ and $(\lambda_3, \mu_3)$ from left to right, respectively.

Figure 6.2 The upper and lower bounds of $\mu_2$
Chapter 7

NUMERICAL RESULTS

This chapter is devoted to presenting some important details of the Fortran implementation of the algorithm from the numerical point of view. Furthermore, we introduce a comparison between our algorithm and two pre-existing competitive algorithms that are used for solving the nonlinear eigenproblems [55, 47]. In addition, we present some numerical results and some characterizations of the test problems that we have solved using our algorithm.

7.1 Fortran Implementation and Stopping Criterion

The nonlinear eigenvalue problem (3.4) is sparse and symmetric positive definite. Our goal is to evaluate a subset of the smallest eigenvalues and their corresponding eigenvectors. The IRL method by D. C. Sorensen and Phong Vu has been determined to be well suited for solving the generalized eigenvalue subproblems that arise in this context. There are some important details of the Fortran implementation of the algorithms that are used. We developed a routine to put the band symmetric positive definite matrix $K$ in the band structure using LA PACK guidelines [29]. The LAPACK routine "DPBFA" is used to factor this band structure matrix. Algorithm 3.3 is implemented in Fortran by three routines: DAXPY from BLAS for matrix vector products (step (b)) and the backward and forward solvers from LAPACK to convert step(a) and step(c), respectively. Reverse communication mechanism (a standard interface over the common statements) was used to:

1. Avoid passing the matrix $A(\lambda)$ through a calling sequence or common statements inside the eigen-solver
2. Construct the input (the three points) for the zero-finder and match these points properly.

3. Link the zero-finder and the eigen-solver and replace some steps by the others inside the zero-finder.

The safeguard is used when the linear system in the rational zero-finder is ill-conditioned. Steps (4) and (5) from algorithm 2.1 are implemented through calls to the Level 2 BLAS routine “DGEMV”. One step of reorthogonalization is carried out at steps (6) of algorithm 2.1 rather than iterating until the test \( \|s\| > \epsilon \|f\| \) is passed. Steps (6.a) and (6.b) were also implemented through calls to DGEMV for more details of the IRL Fortran implementation see [46]. Efficient and portable Fortran software of these algorithms is achieved by relying on the LAPACK and BLAS subroutine libraries. This new technique has been implemented and tested on Sparc Station1 machine, under release 4.1.1 of Sun Unix. The cost of operations were clearly dominated by the matrix vector products and the linear solvers routines. The solution of the nonlinear eigenvalue problem is solutions to a sequence of linear eigenvalue subproblems. So, parallelism may be invoked through the Level 2 BLAS and also through the matrix vector product routines [48].

In practice, the stopping criterion is very sensitive to the problem in hand, so a special care should be taken in its choice. The two common stopping conditions for the nonlinear equations may be stated as follows:

\[
|f(\lambda_{j+1})| \leq \epsilon \quad \text{and} \quad \frac{|\lambda_{j+1} - \lambda_j|}{\max(|\lambda_{j+1}|, |\lambda_j|)} \leq \epsilon
\]  

(7.1)

where, \( \lambda_{j+1} \) is the latest approximated value of \( \lambda \), \( \lambda_j \) is the previous one and \( \epsilon = (\text{macheps})^{\frac{1}{2}} \), where macheps is a very small positive quantity as it is defined in chapter 2. The value \( f(\lambda_{j+1}) \) usually gets small before the step does in any problem for which local convergence is fast. On the other hand, when the convergence is only linear, the step may become small first [18]. The stopping criterion \( |\lambda_{j+1} - \lambda_j| \leq \epsilon \) has the
disadvantage of being machine dependent. The second inequality in (7.1) is scale independent when \( \lambda \neq 0 \). This inequality is a reasonable stopping criterion for the zero-finding techniques, but it will be stringent when \( \lambda_{j+1}, \lambda_j \) are very small, [27]. We used the stopping criterion

\[ |\lambda_{j+1} - \lambda_j| \leq \frac{\epsilon_1}{2} |\lambda_{j+1} + \lambda_j|, \]  

where \( \epsilon_1 = 10^{-6} \) (7.2)

to stop the outer iteration (the zero-finder). The inner iteration (the eigen-solver) stops when \( |\epsilon_k y_j| ||f|| \leq \epsilon \mu_j \), where \( y_j \) is the \( j \)-th Ritz vector corresponding to the Ritz value approximating the required spectrum, \( \mu_j \) is the generalized eigenvalue for the problem \( A(\lambda_j)x = \mu(\lambda_j)x \) and \( \epsilon = 10^{-9} \) is in double precision arithmetic. Stopping this \( k \)-step Lanczos iteration produces a good initial guess for computing the next eigenvalue \( \lambda_{j+1}^* \).

### 7.2 Comparison with Some Existing Techniques

The smallest ten eigenvalues and their corresponding eigenvectors of the test problems are obtained using the safeguarded Lanczos with rational interpolation method developed here, the Lanczos-Based technique with linear interpolation that was introduced in [47] and the determinant search approach that was presented in [55]. The purpose of solving these eigensystems is to illustrate the computational efficiency of the new approach for large scale nonlinear eigensystems and to compare this approach with both the Lanczos-Based procedure and the multiple determinant search technique.

**The multiple determinant parabolic interpolation method** has been developed by Williams and Kennedy [55] for solving the dependent eigenvalue problem (3.3). This technique may be outlined in the following main steps:

1. **Factor** the matrix \([G(\lambda)]\) to find the upper triangular matrix \([U(\lambda)]\).

2. **Evaluate** the determinant of \( G(\lambda) \) " \( \Delta \)"

\[ \Delta(\lambda) = \prod_{i=1}^{n} u_{ii}(\lambda) \]
where, $u_{ii}$ are the diagonal elements of the matrix $U(\lambda)$.

(3) **Determine** a global bounds on the eigenvalue $\lambda_i$, using formula (3.4). Once the global bounds have been established, each eigenvalue $\lambda_i$ can be found in a separate interval $[\lambda_i^l, \lambda_i^u]$ such that,

$$s[G(\lambda^l)] - s[G(\lambda^u)] = 1$$

where, $s[G(\lambda)]$ is the sign count of the upper triangular matrix as it is defined in chapter 3. These bounds, $\lambda_i^l$ and $\lambda_i^u$ can be modified, until we get the uncertainty interval, $[\lambda_i^{ll}, \lambda_i^{lu}]$ for each $\lambda_i$.

(4) **If** the eigenvalues $\lambda_i$ are well separated then

(a) **Construct** a parabola between the three points:

$$(\lambda_i^{ii}, \Delta_i^{ii}), \ (\lambda_i^l, \Delta_i^l) \text{ and } (\lambda_i^u, \Delta_i^u)$$

where, $\Delta_j = \Delta(\lambda_j), \ j = ll, l, u$ and $\lambda_i^l$ is set to be $\frac{\lambda_i^{ll} + \lambda_i^{lu}}{2}$

(b) **Use** the first and the second derivatives, $g$ and $h$ respectively, of the interpolated parabola to write

$$\Delta(\lambda) = \Delta^{ii} + g^{ii}(\lambda - \lambda_i^{ii}) + \frac{1}{2}h(\lambda - \lambda_i^{ii})^2$$

where, $\Delta^{ii}$ and $g^{ii}$ are the values of $\Delta(\lambda)$ and the first derivative at $\lambda_i^{ii}$, respectively.

(c) **Solve** the quadratic equation

$$\Delta^{ii} + g^{ii}(\lambda - \lambda_i^{ii}) + \frac{1}{2}h(\lambda - \lambda_i^{ii})^2 = 0$$

for evaluating the desired eigenvalue, $\lambda$.

(5) **Else** use algorithm 5.1 between $\lambda_i^{ii}$ and $\lambda_i^u$ until convergence.
The use of the combined zero-finding (the quadratic interpolation with bisection method) offers computational advantage over the bisection technique that has been used in a traditional determinant search method [56], especially for higher accuracy requirements. Results from [55] show a 10% improvement in computational measurements over Simpson's method [40]. This technique has some drawbacks:

(1) It is not well-suited for conventional frequency-independent, free vibration analyses.

(2) The evaluation of the determinant of the dynamic stiffness matrix for large nonlinear eigensystems is very expensive, due to the factorization of the entire matrix to evaluate the determinant at each step of the search.

(3) The parameterized determinant curves have a potential erratic behavior as reported in [55].

The Lanczos-Based eigensolution technique evaluates a subset of the natural frequencies and modes from the dependent eigenproblems in structural dynamics. These systems are associated with the mixed finite element modeling procedure which is concerned with the polynomial and frequency dependent displacement fields in formulating the system matrices [47]. This approach is based upon the IRL method for solving the linear eigenvalue subproblems and the secant method for updating the eigenvalue $\lambda$. The Lanczos-Based procedure gave remarkable results but it took 4 function evaluations for each eigenvalue as reported in [47]. In addition, the linear interpolation is a special case of the rational approximation that we used to approximate the generalized eigenvalue curves $\mu_j(\lambda)$. On the other hand, the new technique approximates these parameterized curves by rational functions and it has the ability to update the parameter $\lambda$ twice in each step. Moreover, the independent way of choosing the third point controls the location of the pole and helps to minimize the number of function evaluations.
The results obtained from solving a few test problems with different dimensions are listed in tables 7.1 through 7.8. These test problems were solved by using the new approach, the Lanczos-Based method [47] and the determinant search algorithm [55]. As shown, the safeguarded Lanczos technique gives a better improvement over Lanczos-Based and determinant search techniques in minimizing the number of function evaluations, saving computational time and giving a small function value at the solution. It is clear that the three methods are competitive for the small size problems, but for the large size ones the safeguarded Lanczos method is the winner. In particular, for problems approaching $10^3$ degrees-of-freedom the number of function evaluations by safeguarded Lanczos method is approximately one-half of the number of function evaluations by Lanczos-Based technique. It is also approximately $\frac{1}{20}$ of the number of function evaluations by the determinant search procedure.

Figures 7.1 and 7.2 demonstrate the total time required to find the smallest ten eigenvalues as a function of the problem dimension. Figure 7.1 shows that for problems with dimension less than 100 degrees-of-freedom the Lanczos-Based technique with secant interpolation and the multiple determinant parabolic interpolation approach are competitive with the safeguarded Lanczos nonlinear eigenvalue method using rational interpolation. However, as the problem dimension increases, the safeguarded Lanczos method becomes more efficient as shown in figure 7.2.

### 7.3 Numerical Experiments

The test problems are a frame-type structures involving different unconstrained degrees-of-freedom (sizes between 24 and $10^3$) using the mixed finite element model that contains a frequency independent stiffness matrix, $K$ and a frequency dependent mass matrix $M(\lambda)$, [23, 43]. The mixed model formulation has advantages over the computationally intensive exact finite element model. In the later model the mass and stiffness matrices are dependent on the parameter $\lambda$. Any approach for solving it requires more computational operations for refactorization and redecomposition of
the stiffness dynamic matrix at each step. In addition, the mixed finite element model dominates the conventional finite element model which involves discretization error, [47].

The behavior of the nonlinear functions \( \mu_j(\lambda) \) (the generalized eigenvalues of the positive definite matrix \( A(\lambda) \)) associated with the frame-type systems is monotone increasing on the uncertainty interval. This advantage leads to the result that the parameterized eigenvalue curves are much better behaved than the parameterized determinant curves that are associated with the determinant search method [55].

The linear eigen-solver that we used (MIRL) only requires matrix vector products. It has many other advantages such as maintenance of orthogonality, fixed storage and rapid convergence rate for evaluating a subset of the extremes eigenvalues and their corresponding eigenvectors.

The rational approximations have the advantage over polynomial approximations of any order. They are used to approximate any function with or without a pole. Moreover, they are considered as the generalization of the linear interpolation. The secant method that is based on linear interpolation used to update the eigenvalue \( \lambda \) in the Lanczos-Based technique. In contrast, our zero-finder approach that is based on rational function interpolation is used to approximate the parameterized curves \( \mu_j(\lambda) \) and updates the parameter \( \lambda \). The value of \( \mu_2 \) was computed to be dependent on the physical properties of the system and control the pole.

In all the results reported in this thesis, we have used the exact shifts as described in chapter 2, the values \( k = 10, p = 20 \) in algorithm 6.4, the condition \( |\lambda_{j+1} - \lambda_j| \leq 10^{-6} \) to terminate the multiple determinant search iteration and the inequality (7.2) to terminate the iterations of Lanczos-Based and safeguarded Lanczos approaches. The natural frequencies \( \omega \) are computed by using the formula \( \omega = \frac{\sqrt{\lambda}}{2\pi} \) and their corresponding modes are the eigenvectors. Table 7.9 contains the eigenvectors corresponding to the smallest ten eigenvalues for a problem with dimension 36 degrees-of-freedom.
In the tables 7.1 through 7.8, the safeguarded Lanczos method computes the first eigenvalue by solving at least two linear eigenvalue subproblems because we do not have any information. The computation of each eigenvalue from the rest needs at least solution of one linear eigenproblem. If the generalized eigenvalues \( \mu(\lambda) \) exceed \( 5 \times 10^{-2} \) the step of the safeguarded rational zero-finder is equivalent to one step from secant method (for finding more information for the rational interpolation) and one step from rational zero-finder for updating the parameter \( \lambda \). These features have been reflected in the computations. For example, the evaluation of the first eigenvalue takes one function evaluation less than secant method (see tables 7.1 through 7.8).

**Notations:**

Sr No = the serial number for the eigenvalues,

Cno = the serial number for the components of the eigenvector,

\( \lambda = \) the value of the nonlinear eigenvalue,

t/s = the time in seconds for evaluating the eigenvalue,

err = \( |\lambda_{j+1} - \lambda_j| \),

fe = the number of function evaluations for computing each eigenvalue.

Each function evaluation takes solution of one linear eigenvalue problem (in Lanczos techniques), but it takes one factorization of the dynamic stiffness matrix (in the multiple determinant search approach).
Table 7.1 The smallest ten eigenvalues for a system of size 36 degrees-of-freedom, the half bandwidth of the stiffness matrix = 15 and the number of constraints = 9. In tables 7.1 and 7.2, the safeguarded zero-finder computes the first four eigenvalues without using the safeguard and the safeguard is used for computing the rest because of the values of $\mu(\lambda)$.

Determinant search method gives more accuracy over the Lanczos techniques but it takes several number of the dynamic stiffness matrix factorization.

<table>
<thead>
<tr>
<th>Sr No</th>
<th>Safeguarded with Lanczos</th>
<th>Secant with Lanczos</th>
<th>Determinant search method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\lambda$</td>
<td>$fe$</td>
<td>t/s</td>
</tr>
<tr>
<td>1</td>
<td>0.243</td>
<td>2</td>
<td>0.37</td>
</tr>
<tr>
<td>2</td>
<td>2.460</td>
<td>2</td>
<td>0.37</td>
</tr>
<tr>
<td>3</td>
<td>7.9195</td>
<td>2</td>
<td>0.38</td>
</tr>
<tr>
<td>4</td>
<td>16.314</td>
<td>2</td>
<td>0.43</td>
</tr>
<tr>
<td>5</td>
<td>52.823</td>
<td>6</td>
<td>1.77</td>
</tr>
<tr>
<td>6</td>
<td>71.431</td>
<td>4</td>
<td>1.25</td>
</tr>
<tr>
<td>7</td>
<td>76.624</td>
<td>4</td>
<td>1.32</td>
</tr>
<tr>
<td>8</td>
<td>100.81</td>
<td>4</td>
<td>1.20</td>
</tr>
<tr>
<td>9</td>
<td>101.87</td>
<td>3</td>
<td>1.01</td>
</tr>
<tr>
<td>10</td>
<td>125.50</td>
<td>5</td>
<td>1.64</td>
</tr>
</tbody>
</table>

Table 7.2 The smallest ten eigenvalues for a system of size 60 degrees-of-freedom, the half bandwidth of the stiffness matrix = 25 and the number of constraints = 15.
Table 7.3 The smallest ten eigenvalues for a system of size 90 degrees-of-freedom, the half bandwidth of the stiffness matrix = 25 and the number of constraints = 15. In solving these two systems (7.3 and 7.4) the safeguarded Lanczos technique gives improvement in minimizing the number of function evaluations. For some eigenvalues (for instance the tenth one) the number of function evaluations of safeguarded Lanczos and Lanczos-Based are the same.

Table 7.4 The smallest ten eigenvalues for a system of size 162 degrees-of-freedom, the half bandwidth of the stiffness matrix = 30 and the number of constraints = 18.

<table>
<thead>
<tr>
<th>Sr</th>
<th>Safeguarded with Lanczos</th>
<th>Secant with Lanczos</th>
<th>Determinant search method</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>λ</td>
<td>fe</td>
<td>t/s</td>
</tr>
<tr>
<td>1</td>
<td>0.037</td>
<td>2</td>
<td>7.92</td>
</tr>
<tr>
<td>2</td>
<td>0.348</td>
<td>1</td>
<td>4.19</td>
</tr>
<tr>
<td>3</td>
<td>1.005</td>
<td>1</td>
<td>4.85</td>
</tr>
<tr>
<td>4</td>
<td>2.073</td>
<td>1</td>
<td>6.11</td>
</tr>
<tr>
<td>5</td>
<td>3.629</td>
<td>1</td>
<td>5.70</td>
</tr>
<tr>
<td>6</td>
<td>5.731</td>
<td>1</td>
<td>6.73</td>
</tr>
<tr>
<td>7</td>
<td>8.368</td>
<td>2</td>
<td>11.8</td>
</tr>
<tr>
<td>8</td>
<td>11.38</td>
<td>1</td>
<td>6.59</td>
</tr>
<tr>
<td>10</td>
<td>16.71</td>
<td>1</td>
<td>7.40</td>
</tr>
</tbody>
</table>

Table 7.5 The smallest ten eigenvalues for a system of size 270 degrees-of-freedom, the half bandwidth of the stiffness matrix = 45 and the number of constraints = 27. In these two tables 7.5 and 7.6, the determinant search technique takes a large number of the dynamic stiffness matrix factorization. On the other hand Lanczos techniques take a few number of linear eigenvalue subproblems solutions and give more accuracy.

<table>
<thead>
<tr>
<th>Sr</th>
<th>Safeguarded with Lanczos</th>
<th>Secant with Lanczos</th>
<th>Determinant search method</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>λ</td>
<td>fe</td>
<td>t/s</td>
</tr>
<tr>
<td>1</td>
<td>0.037</td>
<td>2</td>
<td>7.92</td>
</tr>
<tr>
<td>2</td>
<td>0.348</td>
<td>1</td>
<td>4.19</td>
</tr>
<tr>
<td>3</td>
<td>1.005</td>
<td>1</td>
<td>4.85</td>
</tr>
<tr>
<td>4</td>
<td>2.073</td>
<td>1</td>
<td>6.11</td>
</tr>
<tr>
<td>5</td>
<td>3.629</td>
<td>1</td>
<td>5.70</td>
</tr>
<tr>
<td>6</td>
<td>5.731</td>
<td>1</td>
<td>6.73</td>
</tr>
<tr>
<td>7</td>
<td>8.368</td>
<td>2</td>
<td>11.8</td>
</tr>
<tr>
<td>8</td>
<td>11.38</td>
<td>1</td>
<td>6.59</td>
</tr>
<tr>
<td>10</td>
<td>16.71</td>
<td>1</td>
<td>7.40</td>
</tr>
</tbody>
</table>

Table 7.6 The smallest 10 eigenvalues for a system of size 330 degrees-of-freedom, the half bandwidth of the stiffness matrix = 55 and the number of constraints = 33.
<table>
<thead>
<tr>
<th>Sr No</th>
<th>Safeguarded with Lanczos</th>
<th>Secant with Lanczos</th>
<th>Determinant search method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\lambda$</td>
<td>$fe$</td>
<td>$t/s$</td>
</tr>
<tr>
<td>1</td>
<td>0.016</td>
<td>2</td>
<td>14.5</td>
</tr>
<tr>
<td>2</td>
<td>0.152</td>
<td>1</td>
<td>9.19</td>
</tr>
<tr>
<td>3</td>
<td>0.431</td>
<td>1</td>
<td>10.5</td>
</tr>
<tr>
<td>4</td>
<td>0.868</td>
<td>1</td>
<td>11.2</td>
</tr>
<tr>
<td>5</td>
<td>1.484</td>
<td>1</td>
<td>12.5</td>
</tr>
<tr>
<td>6</td>
<td>2.303</td>
<td>1</td>
<td>14.3</td>
</tr>
<tr>
<td>7</td>
<td>3.349</td>
<td>1</td>
<td>15.7</td>
</tr>
<tr>
<td>8</td>
<td>4.641</td>
<td>1</td>
<td>17.9</td>
</tr>
<tr>
<td>9</td>
<td>6.188</td>
<td>1</td>
<td>19.3</td>
</tr>
<tr>
<td>10</td>
<td>7.977</td>
<td>1</td>
<td>19.5</td>
</tr>
</tbody>
</table>

**Table 7.7** The smallest ten eigenvalues for a system of size 585 degrees-of-freedom, the half bandwidth of the stiffness matrix = 65, number of constraints = 39 and $\epsilon_1 = 10^{-5}$ is used in condition (7.2). The results of these two tables (7.7 and 7.8) indicate that the Lanczos techniques are much better than the determinant search procedures for solving the large size problems.

<table>
<thead>
<tr>
<th>Sr No</th>
<th>Safeguarded with Lanczos</th>
<th>Secant with Lanczos</th>
<th>Determinant search method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\lambda$</td>
<td>$fe$</td>
<td>$t/s$</td>
</tr>
<tr>
<td>1</td>
<td>0.009</td>
<td>2</td>
<td>32.5</td>
</tr>
<tr>
<td>2</td>
<td>0.085</td>
<td>1</td>
<td>18.1</td>
</tr>
<tr>
<td>3</td>
<td>0.239</td>
<td>1</td>
<td>21.4</td>
</tr>
<tr>
<td>4</td>
<td>0.477</td>
<td>1</td>
<td>22.4</td>
</tr>
<tr>
<td>5</td>
<td>0.805</td>
<td>1</td>
<td>29.8</td>
</tr>
<tr>
<td>6</td>
<td>1.232</td>
<td>1</td>
<td>29.8</td>
</tr>
<tr>
<td>7</td>
<td>1.769</td>
<td>1</td>
<td>32.2</td>
</tr>
<tr>
<td>8</td>
<td>2.425</td>
<td>1</td>
<td>36.3</td>
</tr>
<tr>
<td>9</td>
<td>3.211</td>
<td>1</td>
<td>39.2</td>
</tr>
<tr>
<td>10</td>
<td>4.136</td>
<td>1</td>
<td>39.2</td>
</tr>
</tbody>
</table>

**Table 7.8** The smallest 10 eigenvalues for a system of size 960 degrees-of-freedom, the half bandwidth of the stiffness matrix = 80 and the number of constraints = 48.
Figure 7.1  The accumulated time for small problems

Figure 7.2  The accumulated time for large problems
<table>
<thead>
<tr>
<th>Cno</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-.032</td>
<td>-.282</td>
<td>0.352</td>
<td>-.253</td>
<td>-.110</td>
<td>-.008</td>
<td>-.019</td>
<td>0.013</td>
<td>0.018</td>
<td>0.151</td>
</tr>
<tr>
<td>2</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>3</td>
<td>0.294</td>
<td>0.017</td>
<td>-.001</td>
<td>-.023</td>
<td>-.140</td>
<td>0.086</td>
<td>-.353</td>
<td>0.175</td>
<td>-.506</td>
<td>0.183</td>
</tr>
<tr>
<td>4</td>
<td>-.032</td>
<td>-.282</td>
<td>0.352</td>
<td>-.253</td>
<td>-.110</td>
<td>-.008</td>
<td>-.019</td>
<td>0.013</td>
<td>0.018</td>
<td>0.151</td>
</tr>
<tr>
<td>5</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>6</td>
<td>-.217</td>
<td>0.009</td>
<td>0.002</td>
<td>-.011</td>
<td>0.132</td>
<td>0.138</td>
<td>0.332</td>
<td>0.227</td>
<td>0.325</td>
<td>0.272</td>
</tr>
<tr>
<td>7</td>
<td>-.032</td>
<td>-.282</td>
<td>0.352</td>
<td>-.253</td>
<td>-.110</td>
<td>-.008</td>
<td>-.019</td>
<td>0.013</td>
<td>0.018</td>
<td>0.151</td>
</tr>
<tr>
<td>8</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>9</td>
<td>0.108</td>
<td>0.017</td>
<td>-.001</td>
<td>-.023</td>
<td>-.139</td>
<td>-.309</td>
<td>-.185</td>
<td>-.524</td>
<td>-.005</td>
<td>0.189</td>
</tr>
<tr>
<td>10</td>
<td>0.032</td>
<td>-.321</td>
<td>-1.11</td>
<td>0.327</td>
<td>0.089</td>
<td>-.040</td>
<td>-.098</td>
<td>-.035</td>
<td>-.049</td>
<td>-.017</td>
</tr>
<tr>
<td>11</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>12</td>
<td>0.334</td>
<td>-.014</td>
<td>0.039</td>
<td>0.005</td>
<td>0.235</td>
<td>-.247</td>
<td>0.371</td>
<td>-.322</td>
<td>0.058</td>
<td>-.238</td>
</tr>
<tr>
<td>13</td>
<td>0.032</td>
<td>-.321</td>
<td>-1.11</td>
<td>0.327</td>
<td>0.089</td>
<td>-.040</td>
<td>-.098</td>
<td>-.035</td>
<td>-.049</td>
<td>-.017</td>
</tr>
<tr>
<td>14</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>15</td>
<td>-.412</td>
<td>-.008</td>
<td>0.018</td>
<td>0.003</td>
<td>-.255</td>
<td>-.121</td>
<td>-.297</td>
<td>0.125</td>
<td>0.202</td>
<td>-.327</td>
</tr>
<tr>
<td>16</td>
<td>0.032</td>
<td>-.321</td>
<td>-1.11</td>
<td>0.327</td>
<td>0.089</td>
<td>-.040</td>
<td>-.098</td>
<td>-.035</td>
<td>-.049</td>
<td>-.017</td>
</tr>
<tr>
<td>17</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>18</td>
<td>0.221</td>
<td>-.014</td>
<td>0.039</td>
<td>0.005</td>
<td>0.233</td>
<td>0.436</td>
<td>0.092</td>
<td>0.168</td>
<td>-.294</td>
<td>-.243</td>
</tr>
<tr>
<td>19</td>
<td>-.012</td>
<td>0.045</td>
<td>-.265</td>
<td>-.309</td>
<td>-.033</td>
<td>0.056</td>
<td>0.138</td>
<td>-.021</td>
<td>-.034</td>
<td>-.125</td>
</tr>
<tr>
<td>20</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>21</td>
<td>0.176</td>
<td>-.029</td>
<td>-.026</td>
<td>0.016</td>
<td>-.299</td>
<td>0.418</td>
<td>-.095</td>
<td>-.066</td>
<td>0.289</td>
<td>0.224</td>
</tr>
<tr>
<td>22</td>
<td>-.012</td>
<td>0.045</td>
<td>-.265</td>
<td>-.309</td>
<td>-.033</td>
<td>0.056</td>
<td>0.138</td>
<td>-.021</td>
<td>-.034</td>
<td>-.125</td>
</tr>
<tr>
<td>23</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>24</td>
<td>-.160</td>
<td>-.018</td>
<td>-.013</td>
<td>-.000</td>
<td>0.340</td>
<td>-.034</td>
<td>-.082</td>
<td>-.175</td>
<td>-.264</td>
<td>0.263</td>
</tr>
<tr>
<td>25</td>
<td>-.012</td>
<td>0.045</td>
<td>-.265</td>
<td>-.309</td>
<td>-.033</td>
<td>0.056</td>
<td>0.138</td>
<td>-.021</td>
<td>-.034</td>
<td>-.125</td>
</tr>
<tr>
<td>26</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>27</td>
<td>0.313</td>
<td>-.029</td>
<td>-.026</td>
<td>0.016</td>
<td>-.297</td>
<td>-.370</td>
<td>0.212</td>
<td>0.294</td>
<td>0.039</td>
<td>0.221</td>
</tr>
<tr>
<td>28</td>
<td>-.030</td>
<td>0.383</td>
<td>0.352</td>
<td>0.251</td>
<td>-.112</td>
<td>-.008</td>
<td>-.019</td>
<td>0.004</td>
<td>0.003</td>
<td>0.303</td>
</tr>
<tr>
<td>29</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>30</td>
<td>0.097</td>
<td>-.014</td>
<td>-.034</td>
<td>-.050</td>
<td>0.345</td>
<td>-.459</td>
<td>-.165</td>
<td>0.528</td>
<td>-.028</td>
<td>-.127</td>
</tr>
<tr>
<td>31</td>
<td>-.030</td>
<td>0.383</td>
<td>0.352</td>
<td>0.251</td>
<td>-.112</td>
<td>-.008</td>
<td>-.019</td>
<td>0.004</td>
<td>0.003</td>
<td>0.303</td>
</tr>
<tr>
<td>32</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>33</td>
<td>-.214</td>
<td>-.007</td>
<td>-.015</td>
<td>-.007</td>
<td>-.366</td>
<td>0.148</td>
<td>0.371</td>
<td>-.191</td>
<td>-.294</td>
<td>-.199</td>
</tr>
<tr>
<td>34</td>
<td>-.030</td>
<td>0.383</td>
<td>0.352</td>
<td>0.251</td>
<td>-.112</td>
<td>-.008</td>
<td>-.019</td>
<td>0.004</td>
<td>0.003</td>
<td>0.303</td>
</tr>
<tr>
<td>35</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>36</td>
<td>0.343</td>
<td>-.014</td>
<td>-.034</td>
<td>-.050</td>
<td>0.343</td>
<td>0.223</td>
<td>-.424</td>
<td>-.226</td>
<td>0.501</td>
<td>-.120</td>
</tr>
</tbody>
</table>

Table 7.9 The eigenvectors corresponding to the smallest 10 eigenvalues for a system of size 36
Chapter 8

CONCLUDING REMARKS

In this research, we have established a numerical algorithm for solving a class of nonlinear eigenproblems and the monotonicity analysis theory for the spectrum of these systems. Our method is based upon the well-known idea linearize, solve and then update. The eigen-solver (IRL) has been reformulated to be well-suited to solve the linear eigenvalue subproblems that arise. For updating the parameter $\lambda$, we have developed a zero-finding technique which is based on the rational interpolation. This type of interpolation can be used to approximate any function of any degree, even when this function has a pole. The third point in this technique has been chosen in a scale independent way. This would guarantee that the pole lies away from the right end of the uncertainty interval. The safeguard has been used to ensure that the successive points are not too close.

The new solution technique is used for approximating frequency-dependent, nonlinear eigensystems associated with exact free vibration analysis in structural dynamics in affiliation with the mixed finite element approach presented in [23]. The mixed finite element scheme employs both the frequency-dependent and polynomial fields in generating the mass and stiffness matrices, respectively. Our algorithm is capable of producing accurate results for large scale problems. Furthermore, we have illustrated its effectiveness in this context through numerical experiments.

The new methodology of the solution developed here is designed to compute a certain set of parameterized nonlinear curves at given values of the parameter "$\lambda$" through the eigen-solver and update this parameter through the safeguarded zero-finder procedure. The IRL approach is a variant of the highly successful implicit $QR$
technique for small dense problems with the Lanczos method for very large symmetric sparse systems. Our method shows that the use of the new eigensolution procedure accurately approximates the exact eigenvalues and the eigenvectors of the nonlinear eigensystems associated with exact vibration analysis. This was illustrated by evaluating the solution for test problems that include the generated problems in [47]. We also compare our results with known published results [55, 47]. Moreover, our approach proves to be competitive in terms of computational efficiency not only for large scale problems but also for small problem sizes. In addition, the new technique proved to be faster than existing methods in computing the smallest ten eigenvalues and their corresponding eigenvectors of these problems.

Our monotonicity analysis strategy demonstrated the behavior of the parameterized eigenvalue curves $\mu_j(\lambda)$ associated with these eigensystems. We have proved that these functions have a monotone increasing behavior which is much better than the behavior of the parameterized curves that are associated with the determinant search techniques. Finally, the safeguarded use of the IRL eigensolution technique is shown to be more computationally efficient than the multiple determinant search technique and Lanczos-based eigenvalue approach. Our numerical approach and our monotonicity analysis theory are sufficiently general that they hold for any problem associated with the mixed finite element formulation that involves a frequency independent stiffness matrix and a frequency dependent mass matrix. The generalized eigenvalues of this problem are monotone increasing on the uncertainty interval.

There are more questions that should be answered:

- The convergence analysis theory of this class of algorithms and the monotonicity analysis for the spectrum of the matrices that are generated by lateral vibrations are still open topics for research.
- We have observed that the functions $\mu_j(\lambda)$ are situated over their tangent planes which is the geometric interpretation for the convex property. The question that arise here is what are the sufficient assumptions that we need to prove the convexity for
these functions theoretically?.

• We proved that the stiffness matrix $K(\lambda)$ is positive definite only on the interval [0,3) for the exact model. The eigensystem of this model will be well-conditioned on the complement of this interval by preconditioning the matrix $K(\lambda)$ or by perturbing its diagonal entries. The questions now are, what is the preconditioner that can do this task? and what is the effect of the perturbations on the eigenvalues of the nonlinear eigensystem?

• We can apply the same technique linearize, solve and then update using rational function approximation for updating the step and a suitable eigen-solver (k-step Arnoldi or subspace iteration) for solving the nonlinear nonsymmetric eigensystems formulated by using the mixed finite element methods.

• Finally, extensions of this idea to other areas of research such as homotopy techniques and linear systems would be an interesting setting of research. The investigations of these topics shall be studied in the future.
Bibliography


[51] P. Swannell. Automatic computation of the natural frequencies of structural
frames using an exact matrix technique. *Theory and Practice in Finite Element


[55] F. W. Williams and D. Kennedy. Reliable use of determinants to solve nonlinear
structural eigenvalue problems efficiently. *International Journal for Numerical

[56] F. W. Williams and W. H. Wittrick. An automatic computational procedure for
calculating natural frequencies of skeletal structures. *International Journal of


[59] Y. F. Zhou and A. Ruhe. Numerical path following and eigenvalue criteria for branch switching. *Large Scale Eigenvalue Problems*, North-Holand,