This is to certify that the

dissertation entitled

Homotopy Methods and Algorithms
for Real Symmetric Eigenproblems

presented by

Kuiyuan Li

has been accepted towards fulfillment
of the requirements for

Ph.D. degree in Mathematics

Tien-Yien Li
Major professor

Date July 18, 1991

HOMOTOPY METHODS AND ALGORITHMS FOR REAL SYMMETRIC EIGENPROBLEMS

By

Kuiyuan Li

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Department of Mathematics

1991

ABSTRACT

HOMOTOPY METHODS AND ALGORITHMS
FOR REAL SYMMETRIC EIGENPROBLEMS

By

Kuiyuan Li

This thesis discusses the homotopy methods and algorithms for real symmetric eigenproblems. It contains two parts. In the first part, a new algorithm is presented for finding all the eigenvalues and the corresponding eigenvectors of a symmetric tridiagonal matrix. The algorithm is based on the homotopy continuation approach coupled with the strategy of 'Divide and Conquer'. Evidenced by the numerical results, the algorithm provides a considerable advance over previous attempts in using homotopy method for symmetric eigenvalue problems. Numerical comparisons of the algorithm with the methods in widely used ESPACK library as well as Cuppen's 'Divide and Conquer' method are presented. It appears that our algorithm is strongly competitive in terms of speed, accuracy and orthogonality and leads in speed in almost all the cases. The performance of the parallel version of the algorithm is also presented.

In the second part, a homotopy method for finding all eigenpairs of a real symmetric matrix pencil (A, B) is given, where A and B are real $n \times n$ symmetric matrices and B is a positive semidefinite or ill-conditioned positive definite matrix. A reduction of pencil (A, B) to pencil $(\tilde{A}, \tilde{B})$ is given, where $\tilde{A}$ is an unreduced symmetric tridiagonal matrix and $\tilde{B}$ is a positive definite diagonal matrix. One can easily forms the eigenpair (x, λ) of pencil (A, B) from the eigenpair (y, λ) of pencil $(\tilde{A}, \tilde{B})$. Furthermore, a formula is presented for finding the number of the finite eigenvalues of pencil (A, B) without actually solving the generalized eigenproblem. By choosing initial pencil properly, the homotopy curves are very well separated and, in general, very flat and easy to follow. The homotopy algorithm is compared with QZ algorithm. The numerical results show that the homotopy algorithm leads in speed in all the cases.

To my daughter Lei.

ACKNOWLEDGMENTS

I would like to thank Professor Tien–Yien Li, my dissertation advisor, for his constant encouragement and support during my graduate study at Michigan State University. I would also like to thank him for suggesting the problems and the helpful directions which made this work possible.

I would like to thank Professor R. O. Hill under whose guidance I took reading course in Symmetric Eigenvalue Problems which laid the foundation for this research.

I would also like to thank my dissertation committee members Professor Q. Du, Professor D. R. Dunninger, Professor R. O. Hill, and Professor D. Yen for their valuable suggestions and their time.

Contents

Part I
The Homotopy Method for Real Symmetric Tridiagonal Eigenproblem

Part II
The Homotopy Method for Real Symmetric Generalized Eigenproblem

List of Tables

List of Figures

PART I

The Homotopy Method for Real Symmetric Tridiagonal Eigenproblem

Chapter 1

The Homotopy Method for Real Symmetric Tridiagonal Eigenproblem

1.1 Introduction

In this chapter, we propose a new algorithm, based on the continuation approach, for finding all the eigenvalues and eigenvectors of a symmetric tridiagonal matrix. Let A be an $n \times n$ real symmetric tridiagonal matrix of the form

$$A = \begin{pmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & O \\ & \ddots & \ddots & \ddots & \\ & & \beta_{n-1} & \alpha_{n-1} & \beta_n \\ O & & & \beta_n & \alpha_n \end{pmatrix}. \tag{1.1}$$

In (1.1), if some $\beta_i = 0$, then $\mathbf{R^n}$ clearly decomposes into two complementary subspaces invariant under A. Thus the eigenproblem decomposes in an obvious way

into two smaller subproblems. Therefore we will assume that each $\beta_i \neq 0$. That is, A is *unreduced*. Our algorithm employs the strategy of 'Divide and Conquer'. First of all, the matrix A is divided into two blocks by letting one of the β_i's equal to zero. Namely, we let

$$D = \begin{pmatrix} D_1 & 0 \\ 0 & D_2 \end{pmatrix} \tag{1.2}$$

where

$$D_1 = \begin{pmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{k-1} & \alpha_{k-1} & \beta_k \\ & & & \beta_k & \alpha_k \end{pmatrix}, D_2 = \begin{pmatrix} \alpha_{k+1} & \beta_{k+2} & & & \\ \beta_{k+2} & \alpha_{k+2} & \beta_{k+3} & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{n-1} & \alpha_{n-1} & \beta_n \\ & & & \beta_n & \alpha_n \end{pmatrix}.$$

We then calculate the eigenvalues of unreduced matrices D_1 and D_2 by using the most efficient algorithm available. Different from Cuppen's 'Divide and Conquer' method [8], our algorithm conquers the matrix A by the homotopy $H : \mathbf{R^n} \times \mathbf{R} \times [0,1] \longrightarrow \mathbf{R^n} \times \mathbf{R}$, defined by

$$\begin{aligned} H(x,\lambda,t) = \; & (1-t)\begin{pmatrix} \lambda x - Dx \\ \underline{x^T x - 1} \\ 2 \end{pmatrix} + t\begin{pmatrix} \lambda x - Ax \\ \underline{x^T x - 1} \\ 2 \end{pmatrix} \\ = & \begin{pmatrix} \lambda x - [(1-t)D + tA]x \\ \underline{x^T x - 1} \\ 2 \end{pmatrix} \\ = & \begin{pmatrix} \lambda x - A(t)x \\ \underline{x^T x - 1} \\ 2 \end{pmatrix} \end{aligned} \tag{1.3}$$

where $A(t) = (1-t)D + tA$ and D is called an *initial matrix*. It can be easily seen that the solution set of $H(x,\lambda,t) = 0$ in (1.3) consists of disjoint smooth curves,

each of which joins an eigenpair of D to one of A [5, 6, 7, 20]. We call each of these curves a *homotopy curve* or an *eigenpath*. Thus, by following the eigenpaths emanating from the eigenpairs of D at $t = 0$, we can reach all the eigenpairs of A at $t = 1$. Theorem 1.1 in next section shows that the eigenvalue component $\lambda(t)$ of each eigenpath $(x(t), \lambda(t))$ is monotonic in t. On the other hand, by Hoffman-Wielandt theorem [31],

$$\sum_{i=1}^{n}(\lambda_i - \lambda_i(t))^2 \leq ||A(t) - A||_F^2 = 2(1-t)^2\beta_{k+1}^2 \quad \text{for any } t \in [0,1] \tag{1.4}$$

where $||.||_F$ is the Frobenius norm and where λ_i and $\lambda_i(t)$ are the i^{th} eigenvalues of A and $A(t)$ respectively. Therefore, all the eigenvalue curves $\lambda_i(t)$ can be quite *flat* if β_{k+1} (=0 in D) is very small, especially when n is very large. As a consequence, the eigenvalue curves $\lambda_i(t)$ are very easy to follow. We shall describe our curve following algorithm in Chapter 2.

The search for fast reliable methods for handling symmetric eigenproblems has produced a number of methods, most notably the QR algorithm, the bisection Sturm sequence method with inverse iteration [14, 15] and the 'Divide and Conquer' method [8, 9, 28]. In Chapter 3, we shall present our numerical results, along with comparisons with these methods. It appears that our algorithm is strongly competitive in terms of speed, accuracy and orthogonality, and leads in speed in almost all the cases.

Modern scientific computing is marked by the advent of vector and parallel computers and the search for algorithms that are to a large extent parallel in nature. A further advantage of our method is that it is to a larger degree parallel, in the sense that each eigenpath is followed independently of the others. This inherent nature of the homotopy method makes the parallel implementation much simpler than the other methods. In Chapter 3, we also show the performance of our parallel algorithm and an indirect comparison with both Divide and Conquer (D&C) [9] and Bisection/Multisections (B/M) [23], which are currently considered the only parallel algorithms available for symmetric eigenproblems. The very high efficiency of our method and its natural parallelism make the algorithm an excellent candidate for a variety of architectures.

Theoretical aspects of the continuation approach to the eigenvalue problems have been studied in [5, 6, 7, 20]. A first attempt was made in [18] to make the method computationally efficient. Its parallel version appeared in [22]. Evidenced by the numerical results, our algorithms given here provide a considerable improvement over the algorithms in [18, 22].

1.2 Preliminary Analysis

Let $(A)^1$ denote the lower $(n-1) \times (n-1)$ submatrix of A , $(A)_1$ denote the upper $(n-1) \times (n-1)$ submatrix of A.

By a straightforward verification, one can prove the following lemma.

Lemma 1.1 *For matrices M and N and a real number c,*

$$det \begin{pmatrix} & & | & \\ & M & | & \\ & & | & c \\ - & -- - & - & - - -- - \\ & & c & | \\ & & | & N \\ & & | & \end{pmatrix} = detM \cdot detN - c^2 det(M)_1 \cdot det(N)^1. \quad (1.5)$$

Let M be a $k \times k$ unreduced symmetric tridiagonal matrix and let $\{ \xi_1 < \xi_2 < ... < \xi_k \}$ and $\{ \eta_1 < \eta_2 < ... < \eta_{k-1} \}$ be the eigenvalues of M and $(M)_1$ respectively. Then, by Cauchy's interlacing theorem [24],

$$\xi_1 < \eta_1 < \xi_2 < ... < \eta_{k-1} < \xi_k. \quad (1.6)$$

For D_1 and D_2 in (1.2), let $f_1 = det(D_1 - \lambda I)$, $f_2 = det(D_2 - \lambda I)$, $f_3 = det((D_1)_1 - \lambda I)$ and $f_4 = det((D_2)^1 - \lambda I)$. Let $(x(t),\lambda(t))$ be an eigenpath of the homotopy $H(x, \lambda, t) = 0$ in (1.3), then for each $0 \leq t \leq 1$, $\lambda(t)$ is an eigenvalue of $A(t) = (1-t)D + tA$ in (1.3), where D is of the form in (1.2).

Theorem 1.1 *If all the eigenvalues of D are distinct then*

i) Either $\lambda(t)$ is constant for all t in [0,1] or strictly monotonic.

ii) $\dot{\lambda}(t)\tilde{\lambda}(t) > 0$ for t small, if $\dot{\lambda}(t) \not\equiv 0$.

Proof: Since

$$det(A(t)-\lambda(t)I) = det \left(\begin{array}{c|c} D_1 - \lambda(t)I & t\beta_{k+1} \\ \hline t\beta_{k+1} & D_2 - \lambda(t)I \end{array} \right) = 0,$$

from (1.5), we have

$$f_1(\lambda(t))f_2(\lambda(t)) - t^2\beta_{k+1}^2 f_3(\lambda(t))f_4(\lambda(t)) = 0. \tag{1.7}$$

If there exists a t_0 in [0,1] for which $f_3(\lambda(t_0))f_4(\lambda(t_0)) = 0$ then either $f_3(\lambda(t_0)) = 0$ or $f_4(\lambda(t_0)) = 0$; say $f_3(\lambda(t_0)) = 0$. It follows from (1.6), $f_1(\lambda(t_0)) \neq 0$. Hence, $f_2(\lambda(t_0)) = 0$ in (1.7); accordingly,

$$f_1(\lambda(t_0))f_2(\lambda(t_0)) - t^2\beta_{k+1}^2 f_3(\lambda(t_0))f_4(\lambda(t_0)) \equiv 0.$$

This implies $det(A(t) - \lambda(t_0)I) = 0$. Thus, $\lambda(t) = \lambda(t_0)$ for all t in [0,1].

Assume $f_3(\lambda(t))f_4(\lambda(t)) \neq 0$ for any t in [0,1]. Write $\dot{\lambda}(t) = \frac{d}{dt}\lambda(t)$. Differentiating (1.7) with respect to t yields,

$$\frac{d}{d\lambda}[f_1(\lambda(t))f_2(\lambda(t)) - t^2\beta_{k+1}^2 f_3(\lambda(t))f_4(\lambda(t))]\dot{\lambda}(t) = 2t\beta_{k+1}^2 f_3(\lambda(t))f_4(\lambda(t)) \tag{1.8}$$

so,

$$\frac{d}{d\lambda}[f_1(\lambda(t))f_2(\lambda(t)) - t^2\beta_{k+1}^2 f_3(\lambda(t))f_4(\lambda(t))] \neq 0 \qquad \text{for any } t \in (0,1].$$

We claim that

$$\frac{d}{d\lambda}[f_1(\lambda(t))f_2(\lambda(t)) - t^2\beta_{k+1}^2 f_3(\lambda(t))f_4(\lambda(t))] \Big|_{t=0} \neq 0. \tag{1.9}$$

For otherwise,

$$\frac{d}{d\lambda}[f_1(\lambda(t))f_2(\lambda(t)) - t^2\beta_{k+1}^2 f_3(\lambda(t))f_4(\lambda(t))]\Big|_{t=0} = \frac{d}{d\lambda}\ [f_1(\lambda(t))f_2(\lambda(t))]\Big|_{t=0}$$

$$= \frac{d}{d\lambda}\ f_1(\lambda(t))\Big|_{t=0} f_2(\lambda(0)) + f_1(\lambda(0))\frac{d}{d\lambda}\ f_2(\lambda(t))\Big|_{t=0} = 0.$$

Since $\lambda(0)$ is an eigenvalue of D, we have $f_1(\lambda(0))f_2(\lambda(0)) = 0$. If $f_1(\lambda(0)) = 0$ then $f_2(\lambda(0)) \neq 0$ since all eigenvalues of D are distinct. Hence, $\frac{d}{d\lambda}f_1(\lambda(t))|_{t=0} = 0$. Consequently, $\lambda(0)$ is a multiple eigenvalue of D_1 which contradicts to the fact that D_1 is unreduced. The proof of (1.9) for the case $f_2(\lambda(0)) = 0$ follows by the same argument.

Now, from (1.8),

$$\dot{\lambda}(t) = \frac{2t\beta_{k+1}^2 f_3(\lambda(t))f_4(\lambda(t))}{\frac{d}{d\lambda}[f_1(\lambda(t))f_2(\lambda(t)) - t^2\beta_{k+1}^2 f_3(\lambda(t))f_4(\lambda(t))]}$$

which implies $\dot{\lambda}(t) \neq 0$ for any $t \in (0,1]$. Obviously, $\dot{\lambda}(t)$ is continuous, hence, $\dot{\lambda}(t)$ is strictly positive or strictly negative. Namely, $\lambda(t)$ is strictly monotonic.

Now let $g = f_1 f_2$, $f = \beta_{k+1}^2 f_3 f_4$ and $h = (g - t^2 f)_\lambda = h(t, \lambda(t))$, then $\dot{\lambda} = 2tf/h$

$$\ddot{\lambda}(t) = -4t^2\frac{f^2(g - t^2 f)_{\lambda\lambda}}{h^3} + 8t^2\frac{f_\lambda f}{h^2} + \frac{2f}{h} \tag{1.10}$$

Since $f^2(g - t^2 f)_{\lambda\lambda}/h^3$ and $f_\lambda f/h^2$ are continuous in [0,1], they are bounded. By (1.10),

$$\ddot{\lambda}(0) = \lim_{t\to 0}\ddot{\lambda}(t) = \lim_{t\to 0} 2f/h$$

Hence $\dot{\lambda}(t)\ddot{\lambda}(t) > 0$ for t small since $\dot{\lambda}(t) = 2tf/h$.

Q.E.D.

Theorem 1.2 (Hoffman and Wielandt [31]) *Let M be an $n \times n$ symmetric matrix. Let $M' \equiv M + E$ where E is a symmetric perturbation of M. Denote the eigenvalues of M by $\{ \xi_1 \leq \xi_2 \leq ... \leq \xi_n \}$, the eigenvalues of M' by $\{ \xi_1' \leq \xi_2' \leq ... \leq \xi_n' \}$, and the eigenvalues of E by $\{ \gamma_1 \leq \gamma_2 \leq ... \leq \gamma_n\}$, then*

$$\sum_{i=1}^{n}(\xi_i - \xi_i')^2 \leq \sum_{i=1}^{n}\gamma_i^2. \tag{1.11}$$

Applying the above theorem on $M = A$ and $M' = A(t) = (1-t)D + tA = A + E$ with

$$E = \begin{pmatrix} & & \vdots & \\ & O & \vdots & \\ & & \vdots & (t-1)\beta_{k+1} \\ \cdots & \cdots & \cdots & \cdots \\ & (t-1)\beta_{k+1} & \vdots & \\ & & \vdots & O \\ & & \vdots & \end{pmatrix}$$

then, (1.11) gives

$$\sum_{i=1}^{n}(\lambda_i - \lambda_i(t))^2 \leq 2(1-t)^2\beta_{k+1}^2 \quad \text{for all } t \text{ in } [0,1] \tag{1.12}$$

where λ_i and $\lambda_i(t)$ are the eigenvalues of A and $A(t)$ respectively.

From (1.12), together with the conclusion in Theorem 1.1 that each $\lambda_i(t)$ is monotonic in t, we see that the smaller β_{k+1} is the *flatter* the eigenvalue curves are, especially when n is very large. To make the eigenvalue curves easy to follow we intend to choose β_{k+1} as small as we can for k in certain range as described in Chapter 2.

Let $A(t) = D + Y(t)$, where $A(t)$, D and $Y(t)$ are real symmetric, and $\lambda_i(t)$, ξ_i and $\mu_i(t)$ be the eigenvalues of $A(t)$, D and $Y(t)$, respectively, with $\lambda_i(t) \leq \lambda_{i+1}(t)$, $\xi_i \leq \xi_{i+1}$ and $\mu_i(t) \leq \mu_{i+1}(t)$, $i = 1, 2, ..., n-1$.

Theorem 1.3 *For any i, j satisfying $1 \leq i + j - 1 \leq n$, and $t \in [0,1]$, the following inequalities hold:*

$$\xi_i + \mu_j(t) \leq \lambda_{i+j-1}(t) \tag{1.13}$$

and

$$\lambda_{n+2-i-j}(t) \leq \xi_{n+1-i} + \mu_{n+1-j}(t). \tag{1.14}$$

Proof: Let $\mathbf{R}_D^{i-1}$, $\mathbf{R}^{i-1}$, and $\mathbf{R}_{Y(t)}^{j-1}$, $\mathbf{R}^{j-1}$ be the subspaces of $\mathbf{R}^n$ defined implicitly by

$$\xi_i = \max_{\mathbf{R}^{i-1}} \min_{s \perp \mathbf{R}^{i-1}} \frac{s^T D s}{s^T s} \equiv \min_{s \perp \mathbf{R}_D^{i-1}} \frac{s^T D s}{s^T s}$$

$$\mu_j(t) = \max_{\mathbf{R}^{j-1}} \min_{s \perp \mathbf{R}^{j-1}} \frac{s^T Y(t) s}{s^T s} \equiv \min_{s \perp \mathbf{R}^{j-1}_{Y(t)}} \frac{s^T Y(t) s}{s^T s}.$$

In fact, $\mathbf{R}^{i-1}_D$ is the span of the eigenvectors corresponding to the $i-1$ smallest eigenvalues of D. Similarly $\mathbf{R}^{j-1}_{Y(t)}$ is the span of the eigenvectors corresponding to the $j-1$ smallest eigenvalues of $Y(t)$. These subspaces may or may not have a nontrivial intersection. Let S be the subspace of the smallest dimension containing both $\mathbf{R}^{i-1}_D$ and $\mathbf{R}^{j-1}_{Y(t)}$. Write $k \equiv dim(S) + 1$. Now,

$$k - 1 = dim(S) \leq (i-1) + (j-1) < n,$$

with equality holds only if the intersection, $\mathbf{R}^{i-1}_D \cap \mathbf{R}^{j-1}_{Y(t)} = \{0\}$.

So,

$$\begin{aligned}
&\lambda_{i+j-1} \geq \lambda_k \\
&= \max_{\mathbf{R}^{k-1}} \min_{x \perp \mathbf{R}^{k-1}} \frac{x^T A(t) x}{x^T x}, \qquad \text{by definition of } \lambda_k, \\
&\geq \min_{x \perp S} \frac{x^T A(t) x}{x^T x}, \qquad \text{since } dim(S) = k-1, \\
&= \min_{x \perp S} \{ \frac{x^T D x}{x^T x} + \frac{x^T Y(t) x}{x^T x} \} \\
&\geq \min_{u \perp S} \frac{u^T D u}{u^T u} + \min_{v \perp S} \frac{v^T Y(t) v}{v^T v} \\
&\geq \min_{u \perp \mathbf{R}^{i-1}_D} \frac{u^T D u}{u^T u} + \min_{v \perp \mathbf{R}^{j-1}_{Y(t)}} \frac{v^T Y(t) v}{v^T v} \\
&\quad \text{(since } \mathbf{R}^{i-1}_D \subseteq S, \text{ and } \mathbf{R}^{i-1}_{Y(t)} \subseteq S,) \\
&= \xi_i + \mu_j.
\end{aligned}$$

Inequality (1.14) can also be proved by following the same line of argument.

Q.E.D.

Let the inertia of $Y(t)$ be (π, ν, ζ), where π, ν and ζ are the number of positive, negative, and zero eigenvalues of $Y(t)$ respectively. Then we have,

Corollary 1.1 $\xi_{k-\nu} \leq \lambda_k(t) \leq \xi_{k+\pi}$, $k = \nu+1, \nu+2, ..., n-\pi$, *for all* $t \in [0,1]$.

Proof: By the definition of ν, $\mu_{\nu+1}(t) \geq 0$. Let $i = k - \nu$, and $j = \nu + 1$ in Theorem 1.3, then

$$\xi_{k-\nu} \leq \xi_{k-\nu} + \mu_{\nu+1}(t) \leq \lambda_k(t).$$

Let $i = n + 1 - k - \pi$, and $j = \pi + 1$ in inequality (1.14), we have

$$\lambda_k(t) \leq \xi_{k+\pi} + \mu_{n-\pi}(t) \leq \xi_{k+\pi}$$

since $\mu_{n-\pi}(t) \leq 0$. Q.E.D.

Let $Y(t) = t(A - D)$, where D is the block diagonal matrix given in (1.2), then

$$Y(t) = \left(\begin{array}{c|c} O & t\beta_{k+1} \\ \hline t\beta_{k+1} & O \end{array}\right).$$

Since $Y(t)$ has exactly one positive eigenvalue and exactly one negative eigenvalue, from Corollary 1.1, the following corollary is immediately achieved.

Corollary 1.2 *For any* $t \in [0, 1]$,

$$\begin{aligned} &\xi_{i-1} \leq \lambda_i(t) \leq \xi_{i+1}, i = 2, 3, ..., n-1, \\ &\lambda_1(t) \leq \xi_1 \\ &\lambda_n(t) \geq \xi_n. \end{aligned}$$

From Theorem 1.1 and Corollary 1.2, the homotopy curve must be one of those in Figure 1.1. Any homotopy curve is bounded by two consecutive dotted lines and no homotopy curve can cross a dotted line.

It is desirable to choose D as a diagonal matrix, consisting of the diagonal part of A, rather than the form in (1.2) as we did in the above analysis. If D is a diagonal matrix, then eigenvalues and corresponding eigenvectors of D are immediately available. Thus, the work of solving the eigenproblem of D is saved. In [18], Li and Rhee showed that this strategy worked very well for certain matrices, such as $[1, i, 1]$, $i = 1, 2, ..., n$. That is, if we choose $D = diag(1, 2, ..., n)$ in solving eigenproblem of

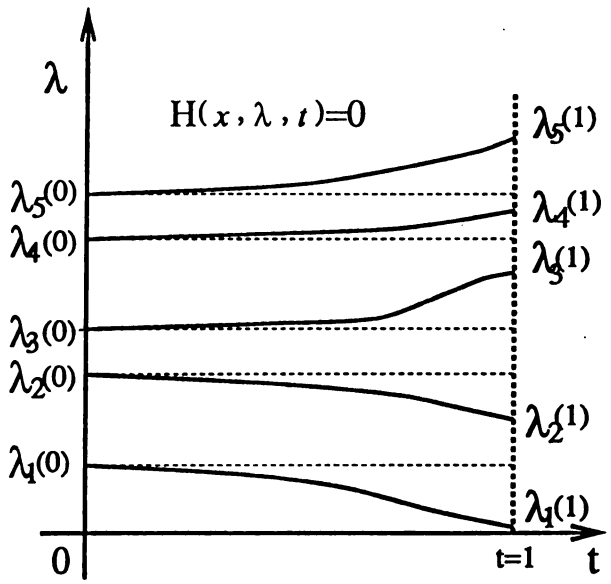

Figure 1.1: Homotopy curves

the matrix $[1, i, 1]$, the eigenpaths are still very flat and easy to follow. However, this strategy breaks down when we solve the eigenproblem of tridiagonal matrices $[1, 2, 1]$. The eigenpaths are rather difficult to follow. In the following analysis, we give some criteria which guarantee the safety of choosing a diagonal starting matrix D.

Note that $(A)^1$ is the lower $(n-1) \times (n-1)$ submatrix of A , $(A)_1$ the upper $(n-1) \times (n-1)$ submatrix of A and $\lambda_i(A)$ is the i^{th} smallest eigenvalue of A.

Lemma 1.2 *If* $\alpha_i < \alpha_{i+1}, i = 1, 2, ..., n-1$ *and if there exists a* c , $0 < c \leq 1$ *such that*

$$(A)^1 - (A)_1 - c \min_{1 \leq i \leq n-1} (\alpha_{i+1} - \alpha_i) I$$

is positive semidefinite then

$$\min_{1 \leq i \leq n-1} (\lambda_{i+1} - \lambda_i) \geq c \min_{1 \leq i \leq n-1} (\alpha_{i+1} - \alpha_i)$$

where $\lambda_i = \lambda_i(A)$.

Proof: Since A is symmetric, so are $(A)_1$ and $(A)^1$.

Let

$$\mu_1 \leq \mu_2 \leq \cdots \leq \mu_{n-1}$$

and

$$\delta_1 \leq \delta_2 \leq \cdots \leq \delta_{n-1}$$

be the eigenvalues of $(A)^1$ and $(A)_1$ respectively, then by Cauchy's interlacing theorem [24],

$$\lambda_1 \leq \mu_1 \leq \lambda_2 \leq \cdots \leq \mu_{n-1} \leq \lambda_n \tag{1.15}$$

$$\lambda_1 \leq \delta_1 \leq \lambda_2 \leq \cdots \leq \delta_{n-1} \leq \lambda_n. \tag{1.16}$$

Since $(A)^1 = (A)_1 + c\alpha I + [(A)^1 - (A)_1 - c\alpha I]$, and $(A)^1 - (A)_1 - c\alpha I$ is positive semidefinite, where

$$\alpha = \min_{1 \leq i \leq n-1} (\alpha_{i+1} - \alpha_i),$$

by the Courant-Fisher maximum characterization [31],

$$\lambda_i((A)^1) \geq \lambda_i((A)_1 + c\alpha I) \qquad \text{for any } i,\ 1 \leq i \leq n-1$$

i.e.,

$$\mu_i - \delta_i \geq c\alpha > 0, \qquad 1 \leq i \leq n-1.$$

By (1.15) and (1.16),

$$\lambda_1 \leq \delta_1 \leq \mu_1 \leq \lambda_2 \leq \cdots \leq \delta_{n-1} \leq \mu_{n-1} \leq \lambda_n.$$

Hence

$$\lambda_{i+1} - \lambda_i \geq \mu_i - \delta_i \geq c\alpha, \qquad 1 \leq i \leq n-1$$

and

$$\min_{1 \leq i \leq n-1} (\lambda_{i+1} - \lambda_i) \geq c \min_{1 \leq i \leq n-1} (\alpha_{i+1} - \alpha_i).$$

Q.E.D.

Corollary 1.3 *If*

$$(A)_1 - \begin{pmatrix} \alpha_1 & & \\ & \ddots & \\ & & \alpha_{n-1} \end{pmatrix} = (A)^1 - \begin{pmatrix} \alpha_2 & & \\ & \ddots & \\ & & \alpha_n \end{pmatrix}.$$

then

$$\min_{1 \leq i \leq n-1} (\lambda_{i+1} - \lambda_i) \geq \min_{1 \leq i \leq n-1} (\alpha_{i+1} - \alpha_i). \tag{1.17}$$

Proof: (1.17) follows immediately from Lemma 1.2, since

$$(A)^1 - (A)_1 - \begin{pmatrix} \alpha_2 - \alpha_1 & & \\ & \ddots & \\ & & \alpha_n - \alpha_{n-1} \end{pmatrix} = 0.$$

Q.E.D.

Let $A(t) = (1-t)D + tA$, where D is a diagonal matrix consisting of the diagonal elements of A, then

Theorem 1.4

$$\min_{1 \le i \le n-1} (\lambda_{i+1}(t) - \lambda_i(t)) \ge c \min_{1 \le i \le n-1} (\alpha_{i+1} - \alpha_i) \qquad t \in [0,1].$$

Proof:

$$\begin{aligned} &(A(t))^1 - (A(t))_1 - \alpha I = t((A)^1 - (A)_1 - \alpha I) \\ &+(1-t)diag(\alpha_2 - \alpha_1 - \alpha, \alpha_3 - \alpha_2 - \alpha, \cdots, \alpha_n - \alpha_{n-1} - \alpha), \end{aligned} \tag{1.18}$$

where

$$\alpha = c \min_{1 \le i \le n-1} (\alpha_{i+1} - \alpha_i), \qquad 0 < c \le 1.$$

Clearly, the second term of the right hand side of (1.18) is positive semidefinite and the first term is positive semidefinite by assumption. Hence, $(A(t))^1 - (A(t))_1 - \alpha I$ is positive semidefinite for $t \in [0,1]$. By Lemma 1.2,

$$\min_{1 \le i \le n-1} (\lambda_{i+1}(t) - \lambda_i(t)) \ge c \min_{1 \le i \le n-1} (\alpha_{i+1} - \alpha_i) \qquad t \in [0,1].$$

Q.E.D.

If A satisfies the conditions in Lemma 1.2, we may choose the initial matrix D as a diagonal matrix consisting of the diagonal elements of A, then, $A(t)$ is an unreduced symmetric tridiagonal matrix and the eigenvalue curves are not only distinct, but also very well separated. There is a lower bound between any two eigenvalue curves so that the eigenvalue curves are easy to follow.

Example 1.1 *A=[1,i,1], where i=1,2,...,20. If we let D=diag { 1,2,...,20 }, then all the eigenvalue curves are very well separated. See Figure 1.2.*

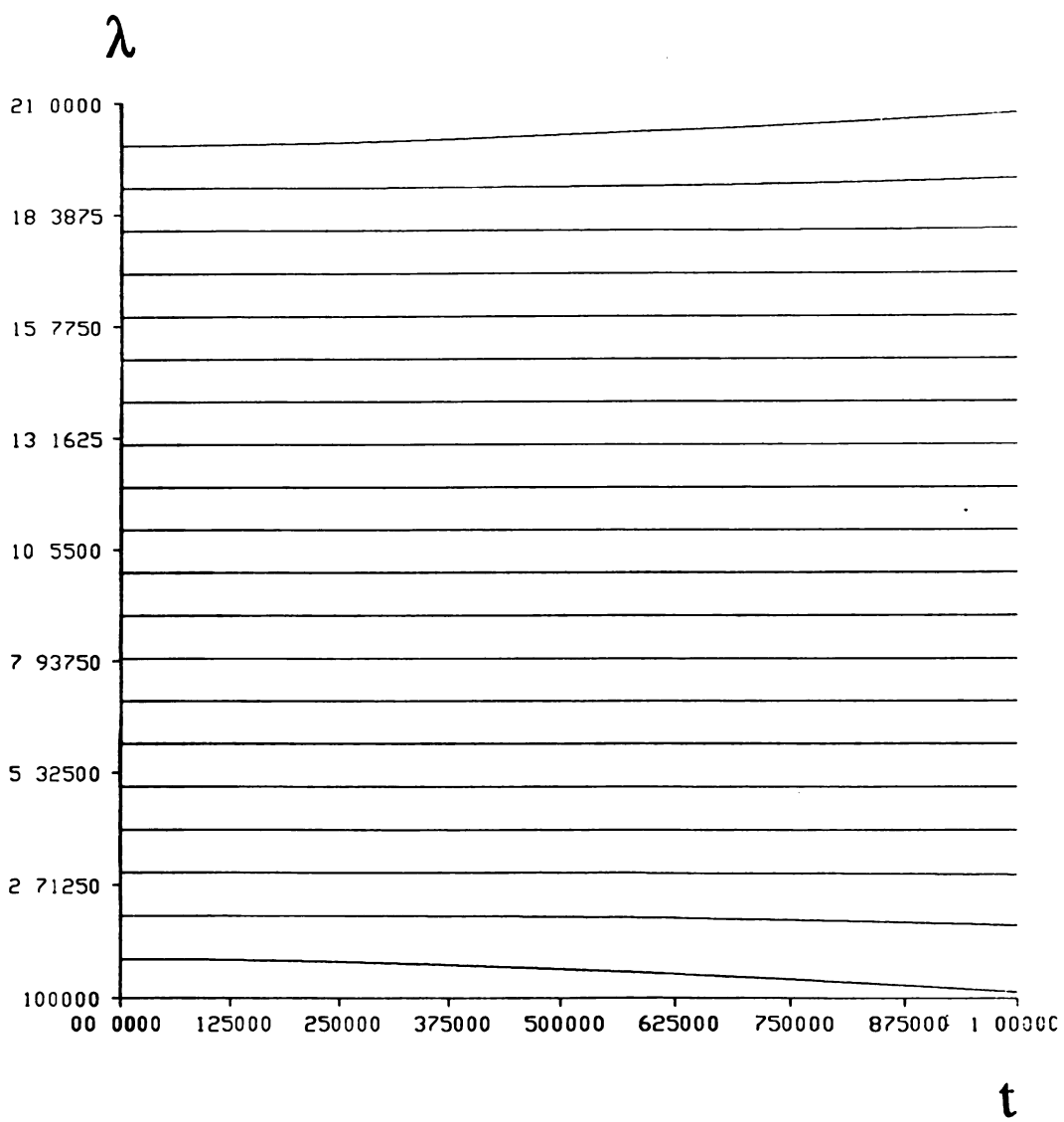

Figure 1.2: The eigenvalue curves of [1,i,1] matrix with D=[0,i,0]

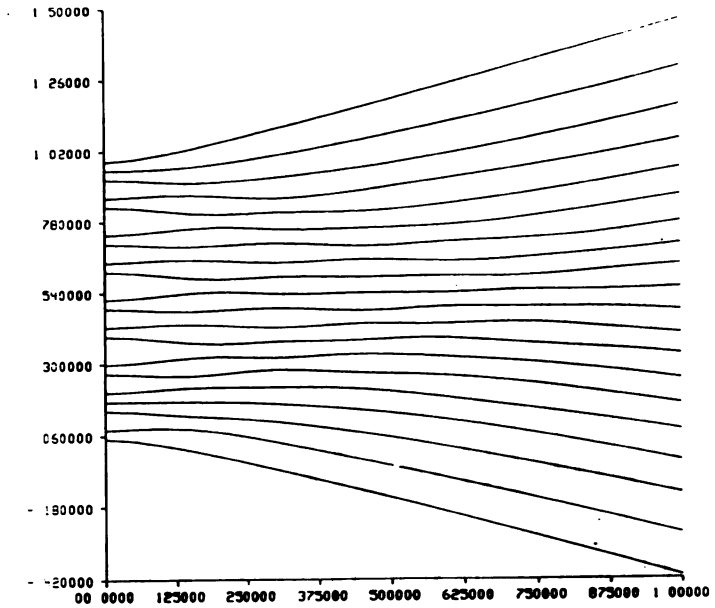

Figure 1.3: The eigenvalue curves of random matrix with $\alpha_i < \alpha_{i+1}$, on diagonal and with D=[0,α_i,0]

Example 1.2 *A is a real symmetric tridiagonal matrix whose diagonal and subdiagonal elements are random numbers between 0 and 1, and whose diagonal elements satisfy* $\alpha_i < \alpha_{i+1}$, *and D is a diagonal matrix consisting of the diagonal elements of A, then all the eigenvalue curves are very well separated. See Figure 1.3.*

Chapter 2

The Homotopy Algorithms for Real Symmetric Tridiagonal Eigenproblem

2.1 Introduction

The basic features of the curve-following scheme of our algorithm to follow the eigenpath $(x(t), \lambda(t))$ are :

(i) Initiating at $t = 0$

(ii) Prediction

(iii) Correction

(iv) Checking

(v) Detection of a cluster and space iteration

(vi) Step-size selection

(vii)Terminating at $t = 1$.

We begin by giving an overview of our algorithm, followed by detailed explanation of these features.

Simple computation shows that Newton's method for the nonlinear problem of

$n+1$ equations

$$F(\lambda, x) = \begin{cases} \lambda x - Ax = 0 \\ \dfrac{x^T x - 1}{2} = 0 \end{cases} \tag{2.1}$$

of the $n+1$ variables λ, $x_1, x_2, ..., x_n$ at $(\lambda^{(n)}, x^{(n)})$ is the inverse iteration,

$$(A - \lambda^{(n)} I)y = x^{(n)}$$

and

$$\begin{cases} \lambda^{(n+1)} = \lambda^{(n)} + \dfrac{(x^{(n)})^T x^{(n)} + 1}{2(x^{(n)})^T y} \\ x^{(n+1)} = (\lambda^{(n+1)} - \lambda^{(n)})y. \end{cases}$$

By making the initial matrix D close to A, the eigenpairs of D should be excellent starting points for applying Newton's method on the eigenproblem (2.1). Based on this observation, our algorithm, in simple terms, is designed to use the homotopy continuation method as a *backup* of Newton's method applied on (2.1). Namely, we solve the eigenvalues of the initial matrix D by using the most efficient method available first, and apply the inverse iteration on (2.1), using each eigenvalue of D as a shift. Then, we switch to Rayleigh quotient iteration (RQI), an inverse iteration with Rayleigh quotient as a shift, to speed up the convergence. This is mainly equivalent to choosing the starting step size $h = 1$ in the usual curve following scheme with zero order prediction and Newton correction to follow the eigenpath $(x(t), \lambda(t))$ of the homotopy $H(x, \lambda, t) = 0$ in (1.3). By checking the Sturm sequence at the convergent point and if this procedure fails to provide the right eigenpair, we shall cut the step size in half. That is, we repeat the process by applying the inverse iteration on

$$\begin{cases} \lambda x - A(t)x = 0 \\ \dfrac{x^T x - 1}{2} = 0 \end{cases}$$

with $t = 0.5$, where $A(t) = (1-t)D + tA$ in (1.3), and then switch to RQI to come back to the right eigenpath $(x(t), \lambda(t))$. Assuming that after i steps, the approximate

value $(x(t_i), \lambda(t_i))$ is known, we always choose the step size $h = 1 - t_i$ at $(x(t_i), \lambda(t_i))$. In this way, we follow the eigenpath from $t = 0$ to $t = 1$.

2.2 Initiating at $t = 0$

As mentioned in Chapter 1, we intend to choose k for which β_{k+1} is as small as possible. To make the sizes of the blocks D_1 and D_2 roughly the same, we limit the choice of k in the range $n/2 - j \leq k \leq n/2 + j$, where j is roughly equal to $n/20$, and find the smallest β_{k+1} by local sorting.

When the initial matrix D is decided, different from the homotopy algorithms in [18, 22] where all the eigenvalues and the eigenvectors of D are calculated in order to start following the eigenpaths, our algorithm only calculates the eigenvalues of D_1 and D_2. These eigenvalues are obtained by using the most efficient method available. We require the accuracy to stay only within one-half or even one-third of the working precision. With this strategy, considerable amount of computing time is reduced.

2.3 Prediction

Assume that after i steps the approximate value $(\tilde{x}(t_i), \tilde{\lambda}(t_i))$ on the eigenpath $(x(t), \lambda(t))$ at t_i is known and the next step-size h is determined; that is, $t_{i+1} = t_i + h$. We want to find an approximate value $(\tilde{x}(t_{i+1}), \tilde{\lambda}(t_{i+1}))$ of $(x(t_{i+1}), \lambda(t_{i+1}))$ on the eigenpath at t_{i+1}. Notice that $(\tilde{x}(t_{i+1}), \tilde{\lambda}(t_{i+1}))$ is an approximate eigenpair of $A(t_{i+1})$. Since $H(x(t), \lambda(t), t) = 0$, we have

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

$$x(t)^T x(t) = 1.$$

Differentiating both equations with respect to t yields,

$$\begin{aligned} (A - D)x(t) + A(t)\dot{x}(t) &= \dot{\lambda}(t)x(t) + \lambda(t)\dot{x}(t) \\ x(t)^T \dot{x}(t) &= 0. \end{aligned} \tag{2.2}$$

For $t = t_i$, multiplying (2.2) on the left by $x^T(t_i)$ yields,

$$\dot{\lambda}(t_i) = x^T(t_i)(A - D)x(t_i) = 2\beta_{k+1}x_k(t_i)x_{k+1}(t_i) \qquad (2.3)$$

where $x(t_i) = (x_1(t_i), ..., x_n(t_i))^T$. In view of (2.3), we use the Euler predictor to predict the eigenvalue at t_{i+1}, namely,

$$\lambda^0(t_{i+1}) = \lambda(t_i) + \dot{\lambda}(t_i)h.$$

It is easy to see that $\dot{\lambda}(0) = 0$ in (2.3). Consequently, $\lambda^0(t_1)$ always equals to $\lambda^0(0)$. To predict the eigenvector, we use the inverse power method on $A(t_{i+1})$ with shift $\lambda^0(t_{i+1})$. That is, we solve

$$(A(t_{i+1}) - \lambda^0(t_{i+1})I)y^0(t_{i+1}) = x(t_i)$$

and let

$$x^0(t_{i+1}) = \frac{y^0(t_{i+1})}{||y^0(t_{i+1})||}.$$

At $t_i = 0$, since we skip the calculations of eigenvectors of D, $x(0)$ is not available. We choose a random vector for $x(0)$.

2.4 Correction

As a corrector, we use the standard RQI, starting with $x^0(t_{i+1})$. To be more precise, at $(x^{j-1}(t_{i+1}), \lambda^{j-1}(t_{i+1}))(j \geq 1)$ let

$$\lambda^j(t_{i+1}) = x^{j-1}(t_{i+1})^T A(t_{i+1})x^{j-1}(t_{i+1})$$

then solve

$$(A(t_{i+1}) - \lambda^j(t_{i+1})I)y^j(t_{i+1}) = x^{j-1}(t_{i+1})$$

and let

$$x^j(t_{i+1}) = \frac{y^j(t_{i+1})}{||y^j(t_{i+1})||}.$$

We repeat the above process to within half of the working precision if single precision is used and one-third of the working precision if double precision is used when $t_{i+1} < 1$, since precision in determining the curve itself is only of secondary interest. We polish $(x^j(t_{i+1}), \lambda^j(t_{i+1}))$ at the end of the path ($t_{i+1} = 1$) by iterating the Rayleigh quotient to machine precision. The stop point $(x^j(t_{i+1}), \lambda^j(t_{i+1}))$ of RQI will be taken as an approximate eigenpair $(\tilde{x}(t_{i+1}), \tilde{\lambda}(t_{i+1}))$ of $A(t_{i+1})$. The cubic convergence rate of RQI makes the corrector highly efficient.

2.5 Checking

When $(\tilde{x}(t_{i+1}), \tilde{\lambda}(t_{i+1}))$ is taken as an approximate eigenpair of $A(t_{i+1})$, the Sturm sequence at $\tilde{\lambda}(t_{i+1}) + \epsilon$ is computed to check that, if we are trying to follow the curve of j^{th} highest eigenvalues, we still on that curve. Here, ϵ is chosen as half of the working precision if single precision is used and one-third of the working precision if double precision is used. If the check fails, we reduce the step size to $h/2$ and repeat the whole process once again beginning with the eigenvalue prediction in Section 2.3.

2.6 Detection of a cluster and subspace iteration

At $t_i = 0$, when all the eigenvalues of D

$$\lambda_1(0) < \lambda_2(0) < ... < \lambda_n(0)$$

are available, we let $\delta = max(10^{-5}, 10^{-2}(\lambda_n(0) - \lambda_1(0))/n)$ if double precision is used (or $\delta = max(10^{-3}, 10^{-2}(\lambda_n(0) - \lambda_1(0))/n)$ if single precision is used). Set λ_i and λ_j in the same group if $|\lambda_i(0) - \lambda_j(0)| < \delta$. If the number of the eigenvalues in any group is bigger than 1, then a cluster is detected. At $t_i \neq 0$, or 1, when $(\tilde{x}(t_i), \tilde{\lambda}(t_i))$ is taken as an approximate eigenpair of $A(t_i)$, after the checking step in Section 2.5, we compute the Sturm sequences at $\tilde{\lambda}(t_i) \pm \delta$ for the purpose of finding the number of eigenvalues of $A(t_i)$ in the interval $(\tilde{\lambda}(t_i) - \delta, \tilde{\lambda}(t_i) + \delta)$. When this number is bigger than 1, a cluster of eigenvalues of $A(t_i)$ is detected.

In those cases, the corresponding eigenvectors are ill-conditioned and this ill-conditioning can cause the inefficiency of the algorithm. To remedy this problem, the inverse power method [24, 31] with $\lambda(t_i)$ as shift is used to construct an approximation of the corresponding eigenspace S of dimension m (= the number of the eigenvalues in the cluster) of $A(t_i)$. This approximate eigenspace S is used as an initial subspace of the subspace iterations at t_{i+1} when we approximate the eigenpairs of $A(t_{i+1})$.

2.7 Step size selection

In the first attempt, we always choose the step size $h = 1 - t_i$ at $t_i < 1$. If after the prediction and the correction steps the checking step fails, we reduce the step size to $h/2$ as mentioned in Section 2.5. This extremely liberal choice of step size can be justified because of the observations we described in the beginning of this Chapter (Section 2.5) as well as the effective checking algorithm. Indeed, since the initial matrix D is chosen to be so close to A, from our experiences, most of the eigenpairs of A can be reached in one step, i.e., $h = 1$.

Very small step size can also cause the inefficiency of the algorithm. Therefore, we impose a minimum γ on the step size h. If $h < \gamma$, we simply give up following the eigenpath and the corresponding eigenpair of A will be calculated at the end of the algorithm by the method of bisection with inverse iterations (see Section 2.8). We usually choose $\gamma \approx 0.25$.

2.8 Terminating at $t = 1$

At $t = 1$, when an approximate eigenvalue $\tilde{\lambda}(1)$ is reached, we compute the Sturm sequence at $\tilde{\lambda}(1) + \epsilon$ with ϵ = machine precision to ensure the correct order. If the checking fails, we have jumped into a wrong eigenpath. More precisely, suppose we are following the i^{th} eigenpair, the checking algorithm detects that we have reached the j^{th} eigenpair instead. In this situation, we will save the j^{th} eigenpair before the

step size is cut. By saving the j^{th} eigenpair, the computation of following the j^{th} eigenpair is no longer needed.

As mentioned in Section 2.7, we may give up following some eigenpaths to avoid adapting a step size that is too small. Without any extra computation, we know exactly which eigenpairs are lost at $t = 1$. In order to find these eigenpairs, we first use the bisection to find the eigenvalues up to the half working precision and then use the inverse iteration and the RQI or subspace iteration (if there are some clusters) to find the eigenpairs.

Chapter 3

Numerical Results of the Homotopy Algorithms for Real Symmetric Tridiagonal Eigenproblem

3.1 Test Matrices

The homotopy continuation algorithm is in its preliminary stage, and much development and testing are necessary. But the numerical results on the examples we have looked at seem remarkable. Our testing matrices are:

(1) The Toeplitz matrix [1, 2, 1], i.e., all its diagonal elements are 2 and off-diagonal elements are 1.

(2) The random matrix with both diagonal and off-diagonal elements being uniformly distributed random numbers between 0 and 1.

(3) The Wilkinson matrix W_n^+. i.e., the matrix $[1, d_i, 1]$, where $d_i = abs((n+1)/2 - i)$, $i = 1, 2, ..., n$ with n odd.

(4) The matrix [1, μ_i, 1], where $\mu_i = i \times 10^{-6}$.

(5) The matrix T_2 : same as matrix [2, 8, 2] except the first diagonal element

$\alpha_1 = 4$.

(6) The glued Wilkinson matrix W_g^+: The matrix consists of j copies of Wilkinson matrices W_k^+ along the diagonal and the off-diagonal elements $\beta_{i,k+1} = 10^{-6}$, where $i = 1, ..., j-1$.

(7) The LAPACK test matrices which include 21 type matrices.

3.2 A serial comparison with the existing methods.

For symmetric tridiagonal matrices, the routine TQL2 in EISPACK [27] implements explicit QL-iteration to find all the eigenpairs. EISPACK also includes a Sturm sequence with inverse iteration method (BISECT+ TINIVT) which is much faster than TQL2. However, it may fail to provide more accurate eigenvectors when the corresponding eigenvalues are very close. A new version by Jessup [15] (B/III) has considerably improved the reliability and the accuracy of the inverse iteration. The 'Divide and Conquer' method for symmetric tridiagonal matrix was suggested by Cuppen [8] and was implemented, combining with a deflation and a robust root finding technique, by Dongarra and Sorenson [9] (TREEQL) (see also [28]).

We shall show the computational results comparing the homotopy continuation method HOMO with those obtained by the methods TQL2, B/III and TREEQL mentioned above. The computations were done on a Sun SPARC station 1.

Table 3.1, 3.2 and 3.3, show the comparisons on the first 6 type test matrices listed in Section 3.1 in terms of speed, accuracy and orthogonality respectively. The homotopy method appears to be strongly competitive in every category and leads in speed by a considerable margin in comparison with all other methods in most of the cases.

Tables 3.4 to 3.9 show the comparisons with (B/III) i.e., DSTEBZ+ DSTEIN, which is the latest code based on bisection with inverse iteration, in terms of speed, accuracy and orthogonality respectively on the LAPACK test matrices. The LAPACK test matrices of type 1 to type 7 are diagonal matrices. Table 3.4 to 3.6 show that

Matrix	Order	Execution Time (second)			
	N	HOMO	B/III	TREEQL	TQL2
$[1,2,1]$	64	2.03	2.43000	5.90999	17.0600
	125	5.88	8.58000	37.6199	114.460
	256	30.25	35.7000	302.859	904.659
	499	100.04	152.920	984.460	2416.14
Random	64	1.13	2.44000	6.09000	17.3200
	125	3.79	8.53000	31.4100	115.880
	256	14.90	34.4500	158.120	949.939
	499	53.19	133.620	235.889	2482.68
W_n^+	65	0.97	1.84000	2.73999	16.1000
	125	3.97	6.21000	8.20001	108.890
	255	16.40	22.5700	31.8398	879.959
	499	57.35	95.5000	57.8300	3869.33
$[1,\mu_i,1]$	64	1.97	2.43000	5.91000	17.1500
	125	6.78	8.64999	37.6500	115.699
	256	30.61	34.1900	303.449	901.810
	499	107.04	129.370	984.410	2424.88
T_2	64	1.80	2.39000	5.80000	16.6800
	125	6.85	8.60000	37.3700	115.000
	256	28.24	34.8600	165.250	939.859
	499	108.46	174.760	979.738	2506.93
W_g^+	64	1.80	1.73000	2.07999	12.2000
	128	8.69	6.07000	6.90000	67.5600
	256	38.85	22.5100	29.0700	490.779
	512	144.05	89.2200	162.040	5303.37

Table 3.1: Execution Time (second) of computed eigenvalues and eigenvectors.

Matrix	Order	$\max_i \|\|Ax_i - \lambda_i x_i\|\|_2/\lambda_{max}$			
	N	HOMO	B/III	TREEQL	TQL2
	64	1.9107D-15	4.9769D-16	1.9027D-15	9.8365D-15
[1, 2, 1]	125	1.9519D-15	8.4237D-16	3.3617D-15	1.0949D-14
	256	2.3533D-15	1.2825D-15	6.0929D-15	2.7538D-14
	499	2.2251D-15	1.7640D-15	7.5172D-15	3.7564D-14
	64	1.9319D-16	3.6661D-16	2.4581D-14	6.0589D-15
Random	125	2.0485D-16	3.5621D-16	5.6541D-14	1.2686D-14
	256	3.9846D-15	3.6238D-16	7.1488D-14	5.3928D-14
	499	1.0330D-13	9.2496D-15	4.8200D-14	5.7588D-14
	65	5.4172D-16	5.4991D-15	1.8801D-13	8.7914D-14
W_n^+	125	3.3967D-16	7.1627D-15	1.0842D-12	3.4443D-13
	255	7.7880D-16	1.4301D-14	1.0868D-12	6.7159D-13
	499	9.1037D-16	2.8363D-14	1.6269D-11	6.5775D-12
	64	4.9149D-15	5.0881D-16	2.4436D-15	8.8198D-15
$[1, \mu_i, 1]$	125	3.2254D-15	7.0876D-16	3.8144D-15	1.1196D-14
	256	4.4093D-15	1.1424D-15	5.4271D-15	2.7578D-14
	499	4.9410D-15	1.7580D-15	8.4456D-15	3.7155D-14
	64	7.1616D-16	1.7737D-15	5.5986D-15	1.5815D-14
T_2	125	7.3060D-16	1.8509D-15	6.5573D-15	2.6714D-14
	256	8.1746D-16	2.2977D-15	1.3413D-14	5.0148D-14
	499	8.0805D-16	3.7362D-15	1.7603D-14	8.1773D-14
	64	8.1580D-16	3.2592D-15	4.3889D-14	3.1906D-14
W_g^+	128	1.2509D-15	1.3014D-14	4.3246D-13	1.1476D-13
	256	2.1321D-15	1.4015D-14	3.8488D-12	6.6732D-13
	512	4.2327D-15	2.3889D-14	3.8490D-12	1.7667D-12

Table 3.2: The residual of computed eigenvalues and eigenvectors.

Matrix	Order	$\max_{i,j} \lvert(X^T X - I)_{i,j}\rvert/\lambda_{max}$			
	N	HOMO	B/III	TREEQL	TQL2
	64	1.7133D-15	9.4041D-15	1.4836D-15	5.9952D-15
[1,2,1]	125	1.3399D-14	3.5176D-14	3.6237D-15	7.5495D-15
	256	1.1554D-14	2.1010D-14	1.4941D-14	1.5987D-14
	499	5.3994D-14	4.1999D-14	1.9569D-14	2.4868D-14
	64	2.3654D-15	2.6645D-15	1.0890D-14	7.3274D-15
Random	125	8.5588D-14	5.5839D-15	4.1540D-14	1.1324D-14
	256	4.0178D-13	3.9968D-15	1.9257D-13	3.6415D-14
	499	1.0330D-13	9.2496D-15	4.8200D-14	4.2410D-14
	65	4.7465D-17	2.2204D-15	1.1102D-15	7.5495D-15
W_n^+	125	1.7693D-17	3.2163D-15	1.2585D-15	1.2656D-14
	255	1.2167D-17	4.3258D-15	4.4408D-15	2.5091D-14
	499	9.3353D-18	1.2278D-13	8.2156D-15	6.9722D-14
	64	3.3790D-15	7.6699D-15	1.0852D-15	4.6629D-15
$[1,\mu_i,1]$	125	1.2375D-14	4.3372D-14	1.2751D-14	6.6613D-15
	256	3.9662D-14	1.6639D-13	8.4026D-15	1.3322D-14
	499	1.8082D-13	1.3981D-13	6.9169D-14	2.4424D-14
	64	5.7249D-16	7.9102D-15	9.3588D-16	5.9952D-15
T_2	125	1.8625D-15	1.3832D-14	3.7492D-15	1.3322D-14
	256	3.2107D-14	1.9904D-14	1.0485D-14	2.1760D-14
	499	6.6773D-14	1.5714D-14	1.9133D-14	4.9293D-14
	64	7.9556D-17	3.5527D-15	6.6613D-16	7.9936D-15
W_g^+	128	6.2610D-16	1.4708D-13	1.1366D-15	9.1038D-15
	256	7.0303D-17	1.0394D-13	4.4408D-15	2.3758D-14
	512	8.4508D-17	1.4259D-13	1.1324D-14	2.6645D-14

Table 3.3: The orthonormality of computed eigenvectors

both algorithms work very well. The matrices of type 9, 17 and 21 have a large cluster with dimension around 4n/5, where n is the order of matrices. On these matrices, while HOMO is not as fast as B/III its accuracy is still competitive. The matrices of type 10 and 18 have a even larger cluster with dimension $n-1$. Although HOMO works, time consuming is out of comparison. Tables 3.4 to 3.9 clearly show that the homotopy method leads in speed by a considerable margin in comparison with B/III on all other types of the LAPACK test matrices.

3.3 An indirect comparison with the existing parallel algorithms

Scientific and engineering research has become increasingly dependent upon the development and implementation of efficient parallel algorithms on modern high - performance computers. Developing algorithms for advanced computers suitable for eigenvalue problems has produced several algorithms, such as Divide and Conquer (D&C)[9] and Bisection/Multisection (B/M)[23] for symmetric tridiagonal matrices.

The homotopy algorithm is to a large degree parallel since each eigenpath can be followed independently. This inherent nature of the homotopy method makes the parallel implementation much simpler than other methods.

In our parallel algorithm, after all the eigenvalues of D are computed and put in the increasing order, we assign each processor to trace roughly n/p eigencurves, where n is the dimension of matrix A and p is the number of the processors being used. Let the first processor trace the first n/P smallest eigencurves from the smallest to the largest and let the second processor trace the second n/p smallest eigencurves, and so on.

We present, in this section, the numerical results of the parallel implementation of our algorithm. All examples were executed on BUTTERFLY GP 1000, a shared memory multiprocessor machine.

Matrix Type	Order N	Execution Time (second)		
		HOMO	B/III	Ratio ((B/III)/HOMO)
Matrix Type 1	10	0.00	0.00	
	32	0.00	9.99999E-03	
	64	2.00005E-02	2.99988E-02	
	128	7.00073E-02	1.00006E-01	
	256	0.320066	0.330017	
	500	1.17969	1.40000	1.18
Matrix Type 2	10	0.00	1.00000E-02	
	32	1.00002E-02	9.99999E-03	
	64	2.00005E-02	2.00005E-02	
	128	8.00018E-02	8.99963E-02	
	256	0.310059	0.350098	
	500	1.18066	1.32996	1.12
Matrix Type 3	10	0.00	9.99999E-03	
	32	0.00	9.99999E-03	
	64	2.00005E-02	3.00026E-02	
	128	6.99921E-02	9.00116E-02	
	256	0.309937	0.340088	
	500	1.17969	1.33997	1.14
Matrix Type 4	10	0.00	0.00	
	32	1.00002E-02	0.00	
	64	2.00005E-02	2.00005E-02	
	128	6.99921E-02	8.99963E-02	
	256	0.320068	0.339966	
	500	1.17969	1.34998	1.14
Matrix Type 5	10	0.00	0.00	
	32	0.00	1.00002E-02	
	64	2.00005E-02	2.99988E-02	
	128	7.99866E-02	9.99908E-02	
	256	0.309937	0.340088	
	500	1.18066	1.35986	1.15
Matrix Type 6	10	0.00	0.00	
	32	1.00002E-02	9.99975E-03	
	64	2.00005E-02	2.00005E-02	
	128	6.99768E-02	9.99908E-02	
	256	0.320068	0.339966	
	500	1.16992	1.32007	1.12
Matrix Type 7	10	0.00	9.99999E-03	
	32	9.99928E-03	1.00002E-02	
	64	2.00005E-02	2.00005E-02	
	128	6.99768E-02	8.99963E-02	
	256	0.320068	0.339966	
	500	1.18066	1.34985	1.14

Table 3.4: Execution Time (second) of computed eigenvalues and eigenvectors from type 1 to 7.

Matrix Type	Order N	$\max_i \|As_i - \lambda_i s_i\|_2/\lambda_{max}$		$\max_{i,j} \|(X^T X - I)_{i,j}\|/\lambda_{max}$	
		HOMO	B/III	HOMO	B/III
Type 1	10	0.0	0.0	0.0	0.0
	32	0.0	0.0	0.0	0.0
	64	0.0	0.0	0.0	0.0
	128	0.0	0.0	0.0	0.0
	256	0.0	0.0	0.0	0.0
	500	0.0	0.0	0.0	0.0
Type 2	10	0.0	0.0	0.0	0.0
	32	0.0	0.0	0.0	0.0
	64	0.0	0.0	0.0	0.0
	128	0.0	0.0	0.0	0.0
	256	0.0	0.0	0.0	0.0
	500	0.0	0.0	0.0	0.0
Type 3	10	0.0	0.0	0.0	0.0
	32	0.0	0.0	0.0	0.0
	64	0.0	0.0	0.0	0.0
	128	0.0	0.0	0.0	0.0
	256	0.0	0.0	0.0	0.0
	500	0.0	0.0	0.0	0.0
Type 4	10	0.0	0.0	0.0	0.0
	32	0.0	0.0	0.0	0.0
	64	0.0	0.0	0.0	0.0
	128	0.0	0.0	0.0	0.0
	256	0.0	0.0	0.0	0.0
	500	0.0	0.0	0.0	0.0
Type 5	10	0.0	0.0	0.0	0.0
	32	0.0	0.0	0.0	0.0
	64	0.0	0.0	0.0	0.0
	128	0.0	0.0	0.0	0.0
	256	0.0	0.0	0.0	0.0
	500	0.0	0.0	0.0	0.0
Type 6	10	0.0	0.0	0.0	0.0
	32	0.0	0.0	0.0	0.0
	64	0.0	0.0	0.0	0.0
	128	0.0	0.0	0.0	0.0
	256	0.0	0.0	0.0	0.0
	500	0.0	0.0	0.0	0.0
Type 7	10	0.0	0.0	0.0	0.0
	32	0.0	0.0	0.0	0.0
	64	0.0	0.0	0.0	0.0
	128	0.0	0.0	0.0	0.0
	256	0.0	0.0	0.0	0.0
	500	0.0	0.0	0.0	0.0

Table 3.5: The residual and the orthonormality of computed eigenvalues and eigenvectors from type 1 to 7.

Matrix Type	Order N	Execution Time (second)		
		HOMO	B/III	Ratio ((B/III)/HOMO)
Matrix Type 8	10	9.00000E-02	1.00000E-01	
	32	0.799999	1.01000	
	64	2.67000	3.99000	
	128	11.3300	15.3600	
	256	36.3000	60.1601	
	500	142.880	226.680	1.59
Matrix Type 9	10	0.11000	1.00000E-01	
	32	1.24000	0.93000	
	64	16.0100	4.54000	
	128	119.060	15.3600	
	256	937.830	154.510	
	500	7334.24	989.060	0.14
Matrix Type 10	10	*	9.00000E-02	
	32	*	0.80000	
	64	*	3.91000	
	128	*	24.2200	
	256	*	171.650	
	500	*	1193.11	*
Matrix Type 11	10	5.99999E-02	0.13000	
	32	0.719999	1.00000	
	64	2.86000	3.86000	
	128	11.1400	24.2800	
	256	46.8599	57.9600	
	500	174.379	217.410	1.25
Matrix Type 12	10	1.00000E-01	0.11000	
	32	0.830001	1.07000	
	64	3.22000	4.17000	
	128	12.2300	14.7800	
	256	40.9399	60.9900	
	500	159.332	229.199	1.44
Matrix Type 13	10	0.110000	0.13000	
	32	0.67000	0.96000	
	64	2.37000	3.84000	
	128	9.95001	14.9800	
	256	33.9600	59.5999	
	500	157.918	223.441	1.42
Matrix Type 14	10	5.00000E-02	0.12000	
	32	0.520000	0.99000	
	64	2.49000	3.82000	
	128	9.70996	14.7700	
	256	41.3799	58.1999	
	500	137.410	217.328	1.58

* The dimension of the subspaces is greater than 4n/5.

Table 3.6: Execution Time (second) of computed eigenvalues and eigenvectors from type 8 to 14.

Matrix Type	Order N	$\max_i \|Ax_i - \lambda_i x_i\|_2/\lambda_{max}$		$\max_{i,j} \|(X^TX - I)_{i,j}\|/\lambda_{max}$	
		HOMO	B/III	HOMO	B/III
Type 8	10	1.11979D-16	1.40169D-16	2.50017D-16	4.44089D-16
	32	2.13068D-16	2.05358D-16	1.39706D-15	1.73556D-15
	64	1.73786D-16	2.97351D-16	3.38698D-15	2.98672D-15
	128	6.41132D-16	2.34456D-16	3.84370D-15	4.58696D-15
	256	5.57906D-16	3.87636D-16	1.05532D-14	1.52117D-14
	500	6.38974D-16	3.31384D-16	2.02084D-14	2.53789D-14
Type 9	10	3.71272D-17	7.55706D-17	8.881784D-16	3.33066D-16
	32	2.17351D-16	1.03313D-16	3.32384D-16	4.44089D-16
	64	6.70107D-17	1.03714D-16	5.10702D-15	4.94924D-16
	128	1.60543D-16	1.40528D-16	5.77315D-15	7.77156D-16
	256	1.14730D-16	2.38822D-16	1.19904D-14	1.21788D-15
	500	1.68505D-16	3.87896D-16	1.77635D-14	2.10941D-15
Type 10	10	*	9.71445D-17	*	3.33066D-16
	32	*	8.51254D-17	*	5.15680D-15
	64	*	1.60196D-16	*	3.72705D-15
	128	*	3.93143D-16	*	3.22413D-14
	256	*	5.81478D-16	*	1.29921D-13
	500	*	6.56717D-17	*	5.22390D-15
Type 11	10	1.02067D-16	1.44292D-16	1.60486D-16	1.49166D-16
	32	1.83393D-16	2.07126D-16	7.75662D-16	1.88355D-15
	64	1.95769D-16	2.35659D-16	1.81188D-15	5.93756D-15
	128	5.08959D-16	2.30321D-16	2.63579D-15	2.81245D-14
	256	6.01778D-16	2.90712D-16	4.65401D-15	1.44048D-13
	500	1.04629D-15	3.44644D-16	1.15582D-14	5.36717D-13
Type 12	10	2.26309D-16	1.84046D-16	3.20972D-16	1.01256D-16
	32	1.76773D-16	2.51273D-16	4.41354D-16	8.44369D-16
	64	6.82696D-16	2.29960D-16	1.55944D-15	4.93778D-16
	128	4.76083D-16	2.51574D-16	4.32069D-15	5.73286D-16
	256	4.63179D-16	2.69293D-16	5.36573D-15	9.84969D-16
	500	1.24891D-15	3.33008D-16	1.46643D-14	6.39731D-15
Type 13	10	1.34514D-16	3.12229D-16	1.17738D-16	1.10379D-16
	32	3.71635D-16	1.69026D-16	1.72823D-16	1.85385D-16
	64	1.81997D-16	2.69087D-16	1.10696D-16	1.79368D-16
	128	8.93746D-16	2.85143D-16	3.04023D-16	4.05102D-16
	256	7.05711D-16	2.21222D-16	1.07909D-15	7.07579D-16
	500	9.36932D-16	2.42164D-16	1.29186D-15	9.48430D-16
Type 14	10	8.47725D-17	2.76210D-16	1.45724D-16	2.96606D-16
	32	1.50639D-16	1.80141D-16	5.42833D-16	1.40457D-16
	64	2.40577D-16	1.91583D-16	1.23016D-15	3.74679D-15
	128	4.24393D-16	2.01754D-16	3.03074D-15	2.02345D-15
	256	2.31166D-16	2.66357D-16	5.47526D-15	8.71391D-15
	500	1.17618D-15	2.81805D-16	1.67426D-14	2.31860D-14

* The dimension of the subspaces is greater than $4n/5$.

Table 3.7: The residual and the orthonormality of computed eigenvalues and eigenvectors from type 8 to 14.

Matrix Type	Order N	Execution Time (second) HOMO	B/III	Ratio ((B/III)/HOMO)
Matrix Type 15	10	0.11600	0.14000	
	32	0.80999	1.08000	
	64	3.42800	4.07999	
	128	9.97003	15.3500	
	256	36.7100	60.3398	
	500	143.809	229.719	1.60
Matrix Type 16	10	8.99997E-02	0.12000	
	32	0.540001	1.03000	
	64	2.34001	3.89000	
	128	9.96997	15.1500	
	256	42.9199	59.5601	
	500	150.508	227.391	1.51
Matrix Type 17	10	9.99999E-02	0.14000	
	32	2.09000	0.970001	
	64	14.4700	4.53001	
	128	105.450	24.5800	
	256	855.330	152.210	
	500	9013.04	998.189	0.11
Matrix Type 18	10	*	9.00002E-02	
	32	*	0.840000	
	64	*	3.85001	
	128	*	24.2900	
	256	*	170.440	
	500	*	1193.35	*
Matrix Type 19	10	0.180000	0.110000	
	32	0.550003	1.02000	
	64	2.39999	3.81001	
	128	9.71997	14.7500	
	256	45.7002	57.5698	
	500	157.070	216.340	1.38
Matrix Type 20	10	8.99997E-02	0.110000	
	32	0.699997	1.04000	
	64	3.73999	3.88000	
	128	19.9099	15.5000	
	256	42.9502	60.2900	
	500	146.160	231.730	1.59
Matrix Type 21	10	7.99999E-02	7.99999E-02	
	32	2.57000	1.01000	
	64	14.1000	4.96001	
	128	142.250	24.4200	
	256	1116.03	150.780	
	500	8702.48	992.301	0.12

* The dimension of the subspaces is greater than 4n/5.

Table 3.8: Execution Time (second) of computed eigenvalues and eigenvectors from type 15 to 21.

Matrix Type	Order N	$\max_i \|\|Ax_i - \lambda_i x_i\|\|_2/\lambda_{max}$		$\max_{i,j} \|(X^T X - I)_{i,j}\|/\lambda_{max}$	
		HOMO	B/III	HOMO	B/III
Type 15	10	7.71083D-17	3.01577D-16	3.64310D-16	1.42399D-16
	32	1.63282D-16	1.84369D-16	1.44074D-15	7.43638D-16
	64	5.46007D-16	2.63286D-16	1.53295D-15	5.86666D-15
	128	1.73512D-16	1.99626D-16	3.77409D-15	7.95686D-15
	256	6.65346D-16	2.16321D-16	6.38724D-15	4.48455D-14
	500	8.55775D-16	3.34659D-16	2.74375D-14	2.08902D-14
Type 16	10	8.96831D-17	1.21381D-16	3.33066D-16	4.44089D-16
	32	7.25319D-17	2.01281D-16	6.66133D-16	7.77156D-16
	64	1.12967D-16	2.25502D-16	1.79116D-15	1.00357D-15
	128	9.94269D-17	2.21255D-16	2.80977D-15	2.27353D-15
	256	4.67682D-16	1.82634D-16	5.41726D-15	5.64456D-15
	500	1.14565D-15	2.37227D-16	1.70815D-13	1.24725D-14
Type 17	10	3.31651D-17	9.61092D-17	1.11022D-15	3.33066D-16
	32	6.46719D-17	1.57507D-16	2.55351D-15	4.44089D-16
	64	8.51895D-17	9.18698D-17	4.21884D-15	1.10557D-15
	128	9.34392D-17	1.84390D-16	7.99360D-15	9.68580D-16
	256	6.45302D-17	1.57202D-16	1.24344D-14	3.00985D-15
	500	2.86221D-16	1.58683D-16	2.06721D-14	3.20500D-15
Type 18	10	*	1.31398D-16	*	7.45834D-16
	32	*	1.07552D-16	*	1.13646D-15
	64	*	8.36246D-17	*	8.05508D-16
	128	*	1.29699D-16	*	1.60028D-15
	256	*	4.47532D-16	*	2.16280D-13
	500	*	3.71442D-16	*	1.32444D-14
Type 19	10	9.40162D-16	5.93782D-17	5.47962D-16	4.44089D-16
	32	7.42684D-17	1.53069D-16	4.85611D-16	7.16000D-16
	64	2.59657D-16	1.25604D-16	7.10580D-16	3.03968D-16
	128	3.18130D-16	1.83999D-16	1.82526D-15	1.14483D-16
	256	8.97197D-16	2.12675D-16	3.58589D-15	4.57344D-15
	500	1.20878D-15	2.19779D-16	7.83319D-15	1.65575D-15
Type 20	10	7.51380D-17	1.09097D-16	2.73991D-16	6.61055D-16
	32	5.79247D-16	1.26621D-16	6.49206D-16	3.09869D-15
	64	1.46996D-16	1.90154D-16	7.30981D-16	3.11117D-15
	128	2.54026D-16	1.65461D-16	1.35331D-15	2.45306D-15
	256	4.45045D-16	2.32804D-16	3.02230D-15	4.26995D-15
	500	6.03169D-16	2.20704D-16	9.48585D-15	3.93531D-15
Type 21	10	5.15682D-17	6.88655D-17	1.17519D-15	3.52558D-16
	32	3.37234D-17	1.00775D-16	2.44274D-15	6.66973D-16
	64	4.33396D-17	1.01234D-16	3.99683D-15	6.66138D-16
	128	5.33451D-17	1.15716D-16	9.54794D-15	6.66135D-16
	256	6.25490D-17	1.14237D-16	1.24344D-14	8.88178D-16
	500	6.04883D-17	1.14603D-16	2.35367D-14	1.75631D-15

* The dimension of the subspaces is greater than $4n/5$.

Table 3.9: The residual and the orthonormality of computed eigenvalues and eigenvectors from type 15 to 21.

The *speed-up* is defined as

$$S_p = \frac{\text{execution time using one processor}}{\text{execution time using } p \text{ processors}}$$

and the *efficiency* is the ratio of the speed-up over p, the number of processors being used.

Table 3.10 shows the execution time as well as the speed-up S_p and the efficiency S_p/p of our algorithm HOMO on matrices [1, 2, 1] and T_2 by using p processors. For the purpose of comparison with other methods, the speed up of our method over TQL2, $\frac{TQL2}{HOMO}$, on [1, 2, 1] are also listed. Similar results on random matrices and Wilkinson matrices are shown on Table 3.11.

We list in Table 3.12, 3.13 and 3.14 some of the results of D&C and B/M. It is somewhat difficult to compare our results with theirs directly since their results were executed on different machines. Using TQL2, an indirect comparison may be obtained. In Table 3.12 (Table 8.3[9]), when eight processors are being used, the speed-up of D&C algorithm(SESUPD) over TQL2 is 9.4 for matrix [1, 2, 1] of order 100 whereas ours is 27.70 for the same matrix of order 125, and the speed-up of D&C algorithm(SESUPD) over TQL2 is 20.00 for order 400 whereas ours is 146.19 for order 499. In Table 3.14 (Table 8.4[9]), when eight processors are being used, the speed-up of D&C algorithm(SESUPD) over TQL2 is 12.1 for random matrix of order 100 whereas ours is 46.16 for the same matrix of order 125 (see Table 3.11), and the speed-up of D&C algorithm(SESUPD) over TQL2 is 60.7 for order 400 whereas ours is 220.00 for order 499.

Table 3.13 (Table 7b[23]) shows the speed-up of B/M (two versions: TREPS1 and TREPS2) and D&C over TQL2 for the matrix $[-1, 2, -1]$ of order 500. It indicates that TQL2/SESUPD = 27.1 and TQL2/TREPS2 = 131.5 whereas TQL2/HOMO = 146.19 for the matrix $[1, 2, 1]$ of order 499 on Table 3.10. This result suggests that the speed-up of the homotopy algorithm is at least as good as TREPS2.

Figure 3.1 shows the efficiency of the matrices [1, 2, 1], T_2, Random matrices and W_n^+ of the order 499 and Figure 3.2 shows the speed-up of the matrices [1, 2, 1], T_2, Random matrices and W_n^+ of the order 499.

[1, 2, 1] matrix							T_2 matrix			
Order N	Nodes p	HOMO (ExeTime)	S_p	$\frac{S_p}{p}$	TQL2 (ExeTime)	$\frac{\text{TQL2}}{\text{HOMO}}$	HOMO (ExeTime)	S_p	$\frac{S_p}{p}$	TQL2 (ExeTime)
	1	7.99	1.0	1.00	27.55	3.44	9.41	1.0	1.00	26.33
65	2	4.51	1.8	0.89		6.10	4.93	1.9	0.95	
	4	2.77	2.9	0.72		9.93	2.88	3.3	0.82	
	1	29.18	1.0	1.00	176.7	6.06	34.98	1.0	1.00	177.33
125	2	15.25	1.9	0.96		11.59	17.55	2.0	1.00	
	4	8.76	3.3	0.83		20.18	9.34	3.7	0.94	
	8	6.38	4.6	0.57		27.70	6.14	5.7	0.71	
	1	122.37	1.0	1.00	1457.	11.91	143.46	1.0	1.00	1497.
	2	69.93	1.9	0.97		20.85	71.76	2.0	1.00	
255	4	33.08	3.7	0.92		44.07	37.2	3.9	0.96	
	8	20.38	6.0	0.75		71.53	22.4	6.4	0.80	
	16	14.28	8.6	0.54		102.08	14.85	9.7	0.60	
	1	477.91	1.0	1.00	10889	22.79	550.90	1.0	1.00	11198
	2	242.01	2.0	0.99		45.00	276.56	2.0	1.00	
499	4	125.45	3.8	0.95		86.80	140.48	3.9	0.98	
	8	74.49	6.4	0.80		146.19	81.72	6.7	0.84	
	16	47.41	10.1	0.63		229.69	52.27	10.5	0.66	

Table 3.10: Execution time (second), speed-up and efficiency of HOMO on [1, 2, 1] and T_2 matrices.

Random matrix							Wilkinson matrix			
Order N	Nodes p	HOMO (ExeTime)	S_p	$\frac{S_p}{p}$	TQL2 (ExeTime)	$\frac{\text{TQL2}}{\text{HOMO}}$	HOMO (ExeTime)	S_p	$\frac{S_p}{p}$	TQL2 (ExeTime)
65	1	6.66	1.0	1.00	29.48	4.43	6.27	1.0	1.00	25.61
	2	3.90	1.7	0.85		7.56	3.35	1.9	0.94	
	4	2.44	2.7	0.68		12.08	2.03	3.0	0.77	
125	1	21.11	1.0	1.00	187.4	8.88	22.10	1.0	1.00	172.05
	2	11.11	1.9	0.95		16.87	11.41	1.9	0.97	
	4	6.04	3.5	0.87		31.03	6.20	3.6	0.89	
	8	4.06	5.2	0.65		46.16	4.26	5.2	0.65	
255	1	80.55	1.0	1.00	1490.	18.50	86.69	1.0	1.00	1367.
	2	41.59	1.9	0.97		35.83	44.27	2.0	0.98	
	4	22.23	3.6	0.91		67.03	23.45	3.7	0.92	
	8	14.35	5.6	0.70		103.83	15.19	5.7	0.71	
	16	10.57	7.6	0.48		140.96	10.55	8.2	0.51	
499	1	301.78	1.0	1.00	11447	37.93	328.24	1.0	1.00	9949.
	2	155.11	1.9	0.97		73.80	166.82	2.0	0.98	
	4	82.47	3.7	0.92		138.80	88.89	3.7	0.92	
	8	52.03	5.8	0.73		220.00	56.16	5.8	0.73	
	16	39.19	7.7	0.48		292.09	40.13	8.2	0.51	

Table 3.11: Execution time (second), speed-up and efficiency of HOMO on Wilkinson and Random matrices.

N	100	200	300	400
TQL2/SESUPD	9.4	15.4	17.7	20.00

Table 3.12: A comparison on the Alliant FX/8 for TQL2 vs. TREEQL on $[1, 2, 1]$ matrices.

i	Algorithm	$\frac{\text{Time(TQL2 on 1 CE)}}{\text{Time}(Alg.\ i \text{ on 8 CEs})}$	$\frac{\text{Time(TQL2 on 8 CE)}}{\text{Time}(Alg.\ i \text{ on 8 CEs})}$
1	TREPS1	32.9	7
2	TREPS2	131.5	28
3	TQL2	4.7	1
4	BISECT+TINVIT	3.6	.8
5	SESUPD	27.1	5.8

Table 3.13: Speed-up over TQL2 on the Alliant FX/8 for computing all the eigenpairs of $[-1, 2, -1]$ matrix of order 500.

N	100	200	300	400
TQL2/SESUPD	12.1	19.5	38.8	60.7

Table 3.14: A comparison on the Alliant FX/8 for TQL2 vs. TREEQL on random matrices.

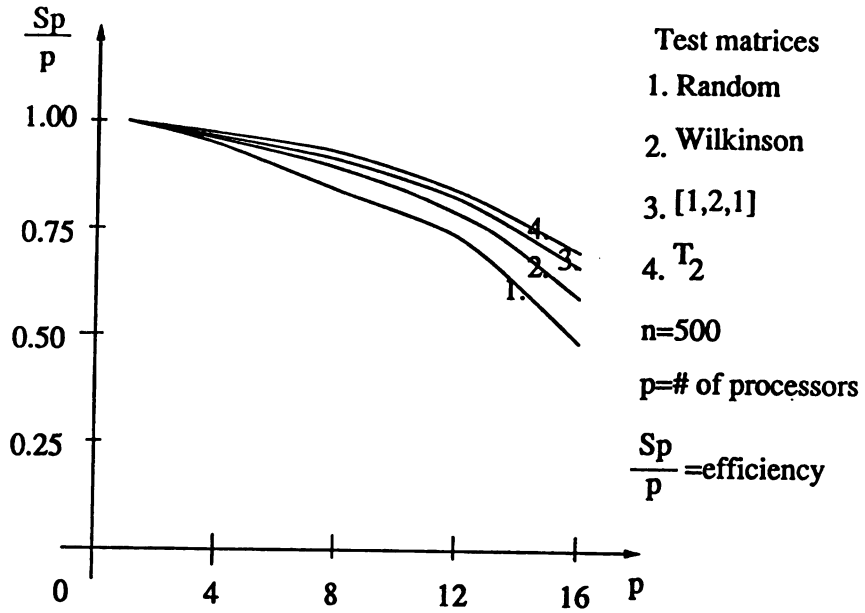

Figure 3.1: Efficiency of homotopy method on [1, 2, 1], T_2, Random matrices and W_n^+

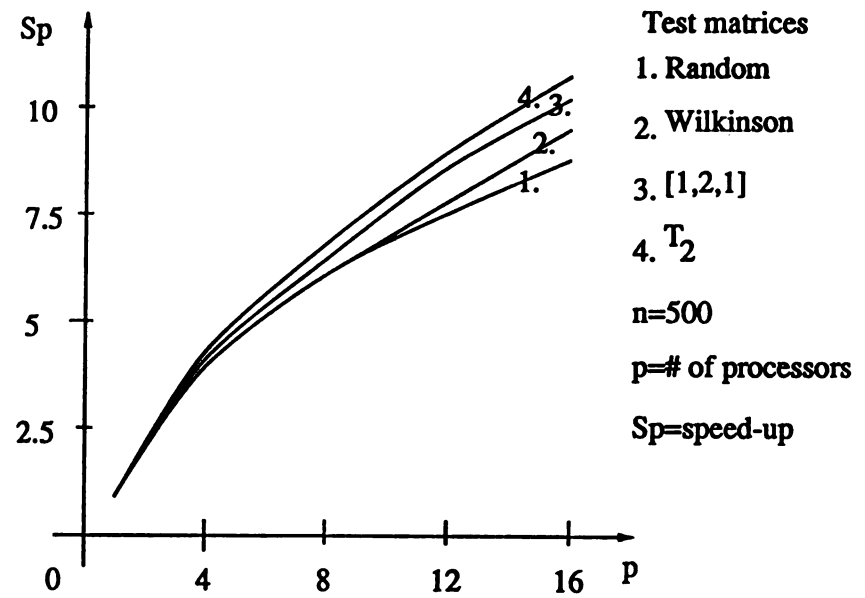

Figure 3.2: Speed-up of homotopy method on [1, 2, 1], T_2, Random matrices and W_n^+

PART II

The Homotopy Method

for

Real Symmetric Generalized Eigenproblem

Chapter 4

Introduction

Consider the symmetric positive semidefinite generalized eigenproblem:

$$Ax = \lambda Bx \tag{4.1}$$

where A and B are $n \times n$ real symmetric matrices, B is positive semidefinite. The pair (A, B) is called a *pencil* of the eigenproblem (4.1), and the eigenpair of (4.1) is called the eigenpair of the pencil (A, B). When B is singular, (4.1) has fewer than n eigenvalues.

Let $X\Delta X^T$ be a spectral decomposition of B, where X is an orthonormal matrix and Δ is a diagonal matrix. Then, (4.1) is equivalent to

$$\tilde{A}y = \lambda \Delta y$$

with $\tilde{A} = X^T A X$ and $y = X^T x$. Hence, we may assume the matrix B in (4.1) is a diagonal matrix with nonnegative diagonal elements. In this case, the MDR reduction [3] can further reduce A to a symmetric tridiagonal matrix and, through the reduction, keep B as a diagonal matrix. Therefore, we will assume hereafter that, in (4.1), A is a symmetric tridiagonal matrix of the form

$$A = \begin{pmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{n-1} & \alpha_{n-1} & \beta_n \\ & & & \beta_n & \alpha_n \end{pmatrix} \tag{4.2}$$

and $B = diag(b_1, b_2, ..., b_n)$ with $b_i \geq 0$.

If $\beta_i = 0$ for some i, $2 \leq i \leq n$, then $\mathbf{R^n}$ can clearly be decomposed into two complementary subspaces invariant under A. Thus the generalized eigenproblem $Ax = \lambda Bx$ is decomposed in an obvious way into two smaller subproblems. Hence, we will assume that each $\beta_i \neq 0$. That is, A is *unreduced*.

In this part of the work, we shall describe our homotopy algorithm for solving eigenpairs of the pencil (A, B) in (4.1). Let D be an $n \times n$ symmetric tridiagonal matrix and consider the homotopy $H : \mathbf{R^n} \times \mathbf{R} \times [0, 1] \longrightarrow \mathbf{R^n} \times \mathbf{R}$, defined by

$$
\begin{aligned}
H(x, \lambda, t) &= (1-t)\begin{pmatrix} \lambda Bx - Dx \\ \underline{x^T Bx - 1} \\ 2 \end{pmatrix} + t\begin{pmatrix} \lambda Bx - Ax \\ \underline{x^T Bx - 1} \\ 2 \end{pmatrix} \\
&= \begin{pmatrix} \lambda Bx - [(1-t)D + tA]x \\ \underline{x^T Bx - 1} \\ 2 \end{pmatrix} \qquad (4.3) \\
&= \begin{pmatrix} \lambda Bx - A(t)x \\ \underline{x^T Bx - 1} \\ 2 \end{pmatrix}
\end{aligned}
$$

where $A(t) = (1-t)D + tA$. The pencil (D, B) is called an *initial pencil*.

In Chapter 5, we shall give a reduction which shows that the eigenvalues of the pencil (A, B) are the same eigenvalues of a pencil $(\tilde{A}, \tilde{B})$ with an unreduced symmetric tridiagonal matrix $\tilde{A}$ and a positive definite diagonal matrix $\tilde{B}$. And, from the reduction, one can easily form the eigenpair (x, λ) of the pencil (A, B) from the eigenpair (y, λ) of the pencil $(\tilde{A}, \tilde{B})$. Therefore, we shall assume the diagonal elements b_i of B are all positive. In such case, we shall prove, in Chapter 6, that the solution set of $H(x, \lambda, t) = 0$ in (4.3) consists of disjoint smooth curves, each of which joins an eigenpair of the pencil (D, B) to an eigenpair of the pencil (A, B). We call each

of these curves a *homotopy curve* or an *eigenpath*. Thus, by following the eigenpaths emanating from the eigenpairs of the pencil (D, B) at $t = 0$, we can reach all the eigenpairs of the pencil (A, B) at $t = 1$.

The algorithm of following an eigenpath will be given in detail in Chapter 7. Some numerical results along with comparison with the QZ method will be presented in Chapter 8. It appears that our algorithm is strongly competitive in terms of speed, accuracy and orthogonality and leads in speed in all the cases.

Chapter 5

Reduction

Let A be an unreduced, symmetric tridiagonal matrix in (4.2) and $b = diag(b_1, b_2,...,b_n)$ with $b_1 > 0$ and $b_i \geq 0$ for $i = 2, 3, ..., n$. In this chapter, we give the details of the reduction of the pencil (A, B) to a pencil $(\tilde{A}, \tilde{B})$, where $\tilde{A}$ is still unreduced tridiagonal and $\tilde{B}$ is positive definite and diagonal.

Lemma 5.1 *Assume*

$$A = \left(\begin{array}{c|c} A_1 & \\ \hline & \beta_{m_1+1} \\ \hline \beta_{m_1+1} & A_1^0 \end{array}\right) \qquad \text{and } B = \begin{pmatrix} B_1 & \\ & O \end{pmatrix},$$

where B_1 is a positive definite diagonal matrix with $dim(B_1) = dim(A_1) = m_1$ and O is the zero matrix with $dim(O) = dim(A_1^0) = n - m_1$. If $det(A_1^0) \neq 0$, then the pencil (A, B) has the same eigenvalues as the eigenproblem:

$$(A_1 + S)y = \lambda B_1 y,$$

where S is a diagonal matrix.

Proof: Clearly, with $x = (x_1, x_2, ..., x_n)^T$, $Ax = \lambda Bx$ is equivalent to

$$A_1 \begin{pmatrix} x_1 \\ \vdots \\ x_{m_1-1} \\ x_{m_1} \end{pmatrix} + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \beta_{m_1+1} x_{m_1+1} \end{pmatrix} = \lambda B_1 \begin{pmatrix} x_1 \\ \vdots \\ x_{m_1-1} \\ x_{m_1} \end{pmatrix} \tag{5.1}$$

and

$$\begin{pmatrix} \beta_{m_1+1} x_{m_1} \\ 0 \\ \vdots \\ 0 \end{pmatrix} + A_1^0 \begin{pmatrix} x_{m_1+1} \\ x_{m_1+2} \\ \vdots \\ x_n \end{pmatrix} = 0. \tag{5.2}$$

Since $det(A_1^0) \neq 0$, $(A_1^0)^{-1}$ exists. From (5.2),

$$\begin{pmatrix} x_{m_1+1} \\ x_{m_1+2} \\ \vdots \\ x_n \end{pmatrix} = -(A_1^0)^{-1} \begin{pmatrix} \beta_{m_1+1} x_{m_1} \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{5.3}$$

Let $(w_1, w_2, ..., w_{n-m_1})^T$ be the solution of

$$A_1^0 \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_{n-m_1} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \tag{5.4}$$

then from (5.3) and (5.4), $x_{m_1+1} = -w_1 \beta_{m_1+1} x_{m_1}$. Let

$$S = \begin{pmatrix} 0 & & O \\ & \ddots & \\ & 0 & \\ O & & -w_1 \beta_{m_1+1}^2 \end{pmatrix} \quad m_1 \times m_1,$$

then (5.1) can be written as:

$$(A_1 + S) \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{m_1} \end{pmatrix} = \lambda B_1 \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{m_1} \end{pmatrix} \tag{5.5}$$

Clearly $A_1 + S$ is an unreduced symmetric tridiagonal matrix and the eigenvalues of (5.5) are exactly the eigenvalues of the pencil (A, B).

Furthermore, if $x = (x_1, x_2, ..., x_{m_1})^T$ is the eigenvector of (5.5) corresponding to the eigenvalue λ, then $x = (x_1, x_2, ..., x_{m_1}, -cw_1, ..., -cw_{n-m_1})^T$, where $c = \beta_{m_1+1} x_{m_1}$, is the eigenvector of the pencil (A, B) corresponding to the same eigenvalue λ.

Q.E.D.

Lemma 5.2 *Assume*

$$A = \left(\begin{array}{c|c|c} A_1 & \beta_{m_1+1} & \\ \hline \beta_{m_1+1} & A_1^0 & \beta_{m_2+1} \\ \hline & \beta_{m_2+1} & A_2 \end{array}\right) \quad \text{and} \quad B = \begin{pmatrix} B_1 & & \\ & O & \\ & & B_2 \end{pmatrix},$$

where $B_i, i = 1, 2$, are positive definite diagonal matrices with $dim(B_1) = dim(A_1) = m_1$ and $dim(B_2) = dim(A_2) = n - m_2$. O is the zero matrix with $dim(O) = dim(A_1^0) = m_2 - m_1$. If $det(A_1^0) \neq 0$, then the pencil (A, B) has the same eigenvalues as the eigenproblem:

$$\left(\begin{pmatrix} A_1 & \\ & A_2 \end{pmatrix} + S\right) y = \lambda \begin{pmatrix} B_1 & \\ & B_2 \end{pmatrix} y, \tag{5.6}$$

where S is a symmetric matrix and the matrix

$$\begin{pmatrix} A_1 & \\ & A_2 \end{pmatrix} + S$$

is unreduced, symmetric and tridiagonal.

Proof: Clearly, $Ax = \lambda Bx$ can be rewritten as:

$$A_1 \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{m_1} \end{pmatrix} + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \beta_{m_1+1} x_{m_1+1} \end{pmatrix} = \lambda B_1 \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{m_1} \end{pmatrix}, \tag{5.7}$$

$$\begin{pmatrix} \beta_{m_1+1}x_{m_1} \\ 0 \\ \vdots \\ 0 \end{pmatrix} + A_1^0 \begin{pmatrix} x_{m_1+1} \\ x_{m_1+2} \\ \vdots \\ x_{m_2} \end{pmatrix} + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \beta_{m_2+1}x_{m_2+1} \end{pmatrix} = 0 \tag{5.8}$$

and

$$\begin{pmatrix} \beta_{m_2+1}x_{m_2} \\ 0 \\ \vdots \\ 0 \end{pmatrix} + A_2 \begin{pmatrix} x_{m_2+1} \\ x_{m_2+2} \\ \vdots \\ x_n \end{pmatrix} = \lambda B_2 \begin{pmatrix} x_{m_2+1} \\ x_{m_2+2} \\ \vdots \\ x_n \end{pmatrix}. \tag{5.9}$$

Since $det(A_1^0) \neq 0$, from (5.8)

$$\begin{pmatrix} x_{m_1+1} \\ x_{m_1+2} \\ \vdots \\ x_{m_2-1} \\ x_{m_2} \end{pmatrix} = -(A_1^0)^{-1} \begin{pmatrix} \beta_{m_1+1}x_{m_1} \\ 0 \\ \vdots \\ 0 \\ \beta_{m_2+1}x_{m_2+1} \end{pmatrix}. \tag{5.10}$$

Let $w = (w_1, ..., w_{m_2-m_1})^T$ and $v = (v_1, ..., v_{m_2-m_1})^T$ be the solutions of

$$A_1^0 w = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \text{ and } A_1^0 v = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix},$$

then, from (5.10),

$$\beta_{m_1+1}x_{m_1+1} = -w_1\beta_{m_1+1}^2 x_{m_1} - v_1\beta_{m_1+1}\beta_{m_2+1}x_{m_2+1}$$

$$\beta_{m_2+1}x_{m_2} = -w_{m_2-m_1}\beta_{m_1+1}\beta_{m_2+1}x_{m_1} - v_{m_2-m_1}\beta_{m_2+1}^2 x_{m_2+1}.$$

Since A_1^0 is unreduced, clearly $v_1 \neq 0$. A_1^0 is symmetric, so is its inverse. Thus, $v_1 = w_{m_2-m_1}$. Let S be an $(n - m_2 + m_1) \times (n - m_2 + m_1)$ matrix of form

$$S = \begin{pmatrix} 0 & & & & & & \\ & \ddots & & & O & & \\ & & 0 & & & & \\ & & & -w_1\beta_{m_1+1}^2 & -v_1\beta_{m_1+1}\beta_{m_2+1} & & \\ & & & -v_1\beta_{m_1+1}\beta_{m_2+1} & -v_{m_2-m_1}\beta_{m_2+1}^2 & & \\ & & O & & & 0 & \\ & & & & & \ddots & \\ & & & & & & 0 \end{pmatrix} \begin{matrix} \\ \\ \\ \text{...}(m_1)^{th}\ row \\ \text{...}(m_1+1)^{th} row \\ \\ \\ \end{matrix} .$$

Then, (5.7) and (5.9) can be rewritten as:

$$\left(\begin{pmatrix} A_1 & \\ & A_2 \end{pmatrix} + S \right) \begin{pmatrix} x_1 \\ \vdots \\ x_{m_1} \\ x_{m_2+1} \\ \vdots \\ x_n \end{pmatrix} = \lambda \begin{pmatrix} B_1 & \\ & B_2 \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_{m_1} \\ x_{m_2+1} \\ \vdots \\ x_n \end{pmatrix} . \qquad (5.11)$$

Clearly, the matrix

$$\begin{pmatrix} A_1 & \\ & A_2 \end{pmatrix} + S$$

is unreduced, symmetric and tridiagonal and the eigenvalues of (5.11) are the same eigenvalues of the pencil (A, B).

Furthermore, if $(x_1, ..., x_{m_1}, x_{m_2+1}, ..., x_n)^T$ is the eigenvector of (5.11) corresponding to the eigenvalue λ, then for $c = \beta_{m_1+1} x_{m_1+1}$ and $d = \beta_{m_2+1} x_{m_2}$,

$$(x_1, ..., x_{m_1}, -w_1 c - v_1 d, ..., -w_{m_2-m_1} c - v_{m_2-m_1} d, x_{m_2+1}, ..., x_n)^T$$

is the eigenvector of the pencil (A, B) corresponding to the same eigenvalue λ.

Q.E.D.

If

$$B = \begin{pmatrix} B_1 & & & & \\ & O_1 & & & \\ & & B_2 & & \\ & & & \ddots & \\ & & & & B_r \end{pmatrix}, \text{ we let } A = \begin{pmatrix} A_1 & * & & & \\ * & A_1^0 & * & & \\ & \ddots & \ddots & \ddots & \\ & & & A_{r-1}^0 & * \\ & & & * & A_r \end{pmatrix}. \quad (5.12)$$

If

$$B = \begin{pmatrix} B_1 & & & & \\ & O_1 & & & \\ & & B_2 & & \\ & & & \ddots & \\ & & & & O_r \end{pmatrix}, \text{ we let } A = \begin{pmatrix} A_1 & * & & & \\ * & A_1^0 & * & & \\ & \ddots & \ddots & \ddots & \\ & & & A_r & * \\ & & & * & A_r^0 \end{pmatrix}. \quad (5.13)$$

where O_i's are zero matrices with $dim(O_i) = dim(A_i^0)$, and B_i's are positive definite diagonal matrices with $dim(B_i) = dim(A_i)$, $i = 1, 2, ..., r$.

Theorem 5.1 *If $det(A_i^0) \neq 0$, $i = 1, 2, ..., r$, then the pencil (A, B) has the same eigenvalues as the eigenproblem:*

$$\left(\begin{pmatrix} A_1 & & & \\ & A_2 & & \\ & & \ddots & \\ & & & A_r \end{pmatrix} + S \right) y = \lambda \begin{pmatrix} B_1 & & & \\ & B_2 & & \\ & & \ddots & \\ & & & B_r \end{pmatrix} y, \quad (5.14)$$

where S is a symmetric matrix and the matrix on the left hand side is an unreduced symmetric tridiagonal matrix.

Proof: By the same line of argument used in Lemma 5.1 and Lemma 5.2, the proof can be immediately achieved by mathematics induction.

Q.E.D.

In the following discussion, let $(A)^1$ and $(A)_1$ denote the lower $(n-1) \times (n-1)$ submatrix and the upper $(n-1) \times (n-1)$ submatrix of A respectively, then $(A)_1^1 = ((A)_1)^1 = ((A)^1)_1$.

Lemma 5.3 *Assume*

$$A = \left(\begin{array}{c|c} A_1 & \begin{matrix} \\ \beta_{m_1+1} \end{matrix} \\ \hline \begin{matrix} \\ \beta_{m_1+1} \end{matrix} & A_1^0 \end{array}\right) \quad \textit{and} \quad B = \begin{pmatrix} B_1 & \\ & O \end{pmatrix},$$

where B_1 is a positive definite diagonal matrix with $dim(B_1) = dim(A_1) = m_1$ and O is a zero matrix with $dim(O) = dim(A_1^0) = n - m_1$. If $det(A_1^0) = 0$, then the pencil (A, B) has the same eigenvalues as the eigenproblem:

$$(A_1)_1 y = \lambda (B_1)_1 y,$$

Proof: Clearly, $Ax = \lambda Bx$ can be rewritten as

$$A_1 \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{m_1} \end{pmatrix} + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \beta_{m_1+1} x_{m_1+1} \end{pmatrix} = \lambda B_1 \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{m_1} \end{pmatrix} \tag{5.15}$$

and

$$\begin{pmatrix} \beta_{m_1+1} x_{m_1} \\ 0 \\ \vdots \\ 0 \end{pmatrix} + A_1^0 \begin{pmatrix} x_{m_1+1} \\ x_{m_1+2} \\ \vdots \\ x_n \end{pmatrix} = 0. \tag{5.16}$$

Since $det(A_1^0) = 0$ and (5.16) has a solution,

$$rank\left(\begin{pmatrix} \beta_{m_1+1} x_{m_1} \\ 0 \\ \vdots \\ 0 \end{pmatrix}, A_1^0 \right) = rank(A_1^0) = n - m_1 - 1.$$

This implies $\beta_{m_1+1}x_{m_1} = 0$, since A_1^0 is unreduced tridiagonal. Since $\beta_{m_1+1} \neq 0$, $x_{m_1} = 0$. Replacing x_{m_1} by 0 in (5.15), we have

$$A_1 \begin{pmatrix} x_1 \\ \vdots \\ x_{m_1-1} \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \beta_{m_1+1}x_{m_1+1} \end{pmatrix} = \lambda B_1 \begin{pmatrix} x_1 \\ \vdots \\ x_{m_1-1} \\ 0 \end{pmatrix}. \tag{5.17}$$

From the last row of (5.17),

$$\beta_{m_1}x_{m_1-1} = -\beta_{m_1+1}x_{m_1+1}. \tag{5.18}$$

Hence,

$$(A_1)_1 \begin{pmatrix} x_1 \\ \vdots \\ x_{m_1-1} \end{pmatrix} = \lambda (B_1)_1 \begin{pmatrix} x_1 \\ \vdots \\ x_{m_1-1} \end{pmatrix}. \tag{5.19}$$

Clearly, the eigenvalues of (5.19) are the same eigenvalues of the pencil (A, B). Moreover, from (5.16) and (5.18),

$$A_1^0 \begin{pmatrix} -\frac{\beta_{m_1}}{\beta_{m_1+1}}x_{m_1-1} \\ x_{m_1+2} \\ \vdots \\ x_n \end{pmatrix} = 0. \tag{5.20}$$

When (5.19) is solved, x_{m_1-1} is known, so (5.20) has a unique solution since $rank(A_1^0) = n - m_1 - 1$. Hence if $(x_1, ..., x_{m_1-1})^T$ is the eigenvector of (5.19) corresponding to the eigenvalue λ, then for $c = -\beta_{m_1}/\beta_{m_1+1}$, $(x_1, ..., x_{m_1-1}, 0, cx_{m_1-1}, x_{m_1+2}, ..., x_n)^T$ is the eigenvector of the pencil (A, B) corresponding to the same eigenvalue λ.

Q.E.D.

Lemma 5.4 *Assume*

$$A = \left(\begin{array}{c|c|c} A_1 & & O \\ \hline & A_1^0 & \\ \hline O & & A_2 \end{array}\right) \quad \begin{matrix} \beta_{m_1+1} \\ \beta_{m_1+1} \end{matrix} \quad \begin{matrix} \beta_{m_2+1} \\ \beta_{m_2+1} \end{matrix} \quad \text{and } B = \begin{pmatrix} B_1 & & \\ & O & \\ & & B_2 \end{pmatrix},$$

where $B_i, i = 1, 2$, are positive definite diagonal matrices with $dim(B_1) = dim(A_1) = m_1$ and $dim(B_2) = dim(A_2) = n - m_2$. O is a zero matrix with $dim(O) = dim(A_1^0) = m_2 - m_1$. If $det(A_1^0) = 0$, then the pencil (A, B) has the same eigenvalues as the eigenproblem:

$$\left(\begin{pmatrix} (A_1)_1 & & \\ & 0 & \\ & & (A_2)^1 \end{pmatrix} + S\right) y = \lambda \left(\begin{pmatrix} (B_1)_1 & & \\ & 0 & \\ & & (B_2)^1 \end{pmatrix} + T\right) y, \tag{5.21}$$

where S is a symmetric matrix and the matrix on the left hand side in (5.21) is an unreduced symmetric tridiagonal matrix. T is a diagonal matrix and the matrix on the right hand side is positive definite and diagonal.

Proof: If $m_2 - m_1 = 1$, i.e., A_1^0 is an 1×1 matrix, then $A_1^0 = 0$ since $det(A_1^0) = 0$. Hence, $Ax = \lambda Bx$ can be rewritten as

$$A_1 \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{m_1} \end{pmatrix} + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \beta_{m_1+1} x_{m_1+1} \end{pmatrix} = \lambda B_1 \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{m_1} \end{pmatrix},$$

$$\beta_{m_1+1} x_{m_1} + \beta_{m_2+1} x_{m_2+1} = 0 \tag{5.22}$$

and

$$\begin{pmatrix} \beta_{m_2+1}x_{m_2} \\ 0 \\ \vdots \\ 0 \end{pmatrix} + A_2 \begin{pmatrix} x_{m_2+1} \\ x_{m_2+2} \\ \vdots \\ x_n \end{pmatrix} = \lambda B_2 \begin{pmatrix} x_{m_2+1} \\ x_{m_2+2} \\ \vdots \\ x_n \end{pmatrix}.$$

Notice that $x_{m_2} = x_{m_1+1}$, and B_1 and B_2 are positive definite. We may rewrite (5.22) as:

$$(A_1)_1 \begin{pmatrix} x_1 \\ \vdots \\ x_{m_1-2} \\ x_{m_1-1} \end{pmatrix} + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ -\frac{\beta_{m_1}\beta_{m_2+1}}{\beta_{m_1+1}}x_{m_2+1} \end{pmatrix} = \lambda(B_1)_1 \begin{pmatrix} x_1 \\ \vdots \\ x_{m_1-2} \\ x_{m_1-1} \end{pmatrix},$$

$$-\beta_{m_1}x_{m_1-1} + \frac{\alpha_{m_1}\beta_{m_2+1}}{\beta_{m_1+1}}x_{m_2+1} - \beta_{m_1+1}x_{m_2} = \lambda\frac{b_{m_1}\beta_{m_2+1}}{\beta_{m_1+1}}x_{m_2+1} \tag{5.23}$$

and

$$\beta_{m_2+1}x_{m_2} + \alpha_{m_2+1}x_{m_2+1} + \beta_{m_2+2}x_{m_2+2} = \lambda b_{m_2+1}x_{m_2+1} \tag{5.24}$$

$$\begin{pmatrix} \beta_{m_2+2}x_{m_2+1} \\ 0 \\ \vdots \\ 0 \end{pmatrix} + (A_2)^1 \begin{pmatrix} x_{m_2+2} \\ x_{m_2+3} \\ \vdots \\ x_n \end{pmatrix} = \lambda(B_2)^1 \begin{pmatrix} x_{m_2+2} \\ x_{m_2+3} \\ \vdots \\ x_n \end{pmatrix}.$$

Solving for x_{m_2} in (5.23) and substituting it into (5.24), yields

$$-\frac{\beta_{m_1}\beta_{m_2+1}}{\beta_{m_1+1}}x_{m_1-1} + (\alpha_{m_1}\frac{\beta^2_{m_2+1}}{\beta^2_{m_1+1}} + \alpha_{m_2+1})x_{m_2+1} + \beta_{m_2+2}x_{m_2+2}$$

$$= \lambda(b_{m_2+1} + \frac{\beta^2_{m_2+1}}{\beta^2_{m_1+1}}b_{m_1})x_{m_2+1}.$$

Let S and T be $(n - m_2 + m_1 - 1) \times (n - m_2 + m_1 - 1)$ matrices of forms.

$$S = \begin{pmatrix} 0 & & & & & \\ & \ddots & & & O & \\ & & 0 & -\frac{\beta_{m_1}\beta_{m_2+1}}{\beta_{m_1+1}} & & \\ & & -\frac{\beta_{m_1}\beta_{m_2+1}}{\beta_{m_1+1}} & \alpha_{m_1}\frac{\beta^2_{m_2+1}}{\beta^2_{m_1+1}} + \alpha_{m_2+1} & \beta_{m_2+2} & \\ & & & \beta_{m_2+2} & 0 & \\ & O & & & & \ddots \\ & & & & & 0 \end{pmatrix} \begin{matrix} \\ \\ ...(m_1-1)^{th}row \\ ...(m_1)^{th}\ row \\ ...(m_1+1)^{th}row \\ \\ \\ \end{matrix}$$

and

$$T = \begin{pmatrix} 0 & & & & & \\ & \ddots & & & O & \\ & & 0 & & & \\ & & & b_{m_2+1} + \frac{\beta^2_{m_2+1}}{\beta^2_{m_1+1}} b_{m_1} & & \\ & & & & 0 & \\ & O & & & & \ddots \\ & & & & & 0 \end{pmatrix} \begin{matrix} \\ \\ \\ (m_1)^{th}row \\ \\ \\ \\ \end{matrix} .$$

Then

$$\left(\begin{pmatrix} (A_1)_1 & & \\ & 0 & \\ & & (A_2)^1 \end{pmatrix} + S \right) \begin{pmatrix} x_1 \\ \vdots \\ x_{m_1-1} \\ x_{m_2+1} \\ \vdots \\ x_n \end{pmatrix} = \lambda \left(\begin{pmatrix} (B_1)_1 & & \\ & 0 & \\ & & (B_2)^1 \end{pmatrix} + T \right) \begin{pmatrix} x_1 \\ \vdots \\ x_{m_1-1} \\ x_{m_2+1} \\ \vdots \\ x_n \end{pmatrix} . \tag{5.25}$$

Clearly,

$$b_{m_2+1} + \frac{\beta^2_{m_2+1}}{\beta^2_{m_1+1}} b_{m_1} > 0,$$

and the matrix on the left hand side in (5.25) is unreduced symmetric tridiagonal. The eigenvalues of (5.25) are the same eigenvalues of the pencil (A, B).

Furthermore, after solving (5.25), x_{m_1} and x_{m_2} can be obtained from (5.22) and from (5.23). Hence if $(x_1, ..., x_{m_1-1}, x_{m_2+1}, ..., x_n)^T$ is the eigenvector of (5.25) corresponding to the eigenvalue λ, then $(x_1, ..., x_{m_1-1}, x_{m_1}, x_{m_2}, x_{m_2+1}, ..., x_n)^T$ is the eigenvector of the pencil (A, B) corresponding to the same eigenvalue λ.

If A_1^0 is $(m_2 - m_1) \times (m_2 - m_1)$, where $m_2 - m_1 \geq 2$, then $Ax = \lambda Bx$ can be written as:

$$(A_1)_1 \begin{pmatrix} x_1 \\ \vdots \\ x_{m_1-2} \\ x_{m_1-1} \end{pmatrix} + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \beta_{m_1} x_{m_1} \end{pmatrix} = \lambda (B_1)_1 \begin{pmatrix} x_1 \\ \vdots \\ x_{m_1-2} \\ x_{m_1-1} \end{pmatrix}, \tag{5.26}$$

$$\beta_{m_1} x_{m_1-1} + \alpha_{m_1} x_{m_1} + \beta_{m_1+1} x_{m_1+1} = \lambda b_{m_1} x_{m_1}, \tag{5.27}$$

$$\begin{pmatrix} \beta_{m_1+1} x_{m_1} \\ 0 \\ \vdots \\ 0 \end{pmatrix} + A_1^0 \begin{pmatrix} x_{m_1+1} \\ x_{m_1+2} \\ \vdots \\ x_{m_2} \end{pmatrix} + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \beta_{m_2+1} x_{m_2+1} \end{pmatrix} = 0, \tag{5.28}$$

$$\beta_{m_2+1} x_{m_2} + \alpha_{m_2+1} x_{m_2+1} + \beta_{m_2+2} x_{m_2+2} = \lambda b_{m_2+1} x_{m_2+1}, \tag{5.29}$$

and

$$\begin{pmatrix} \beta_{m_2+2} x_{m_2+1} \\ 0 \\ \vdots \\ 0 \end{pmatrix} + (A_2)^1 \begin{pmatrix} x_{m_2+2} \\ x_{m_2+3} \\ \vdots \\ x_n \end{pmatrix} = \lambda (B_2)^1 \begin{pmatrix} x_{m_2+2} \\ x_{m_2+3} \\ \vdots \\ x_n \end{pmatrix}. \tag{5.30}$$

Since $rank(A_1^0) = m_2 - m_1 - 1$ and A_1^0 is unreduced tridiagonal,

$$A_1^0 \begin{pmatrix} 1 \\ y_1 \\ \vdots \\ y_{m_2-m_1-1} \end{pmatrix} = 0 \tag{5.31}$$

has a unique solution. Multiplying $(1, y_1, ..., y_{m_2-m_1-1})$ on the left in (5.28), yields $\beta_{m_1+1}x_{m_1} = -y_{m_2-m_1-1}\beta_{m_2+1}x_{m_2+1}$.

Now (5.26) and (5.27) can be rewritten as

$$(A_1)_1 \begin{pmatrix} x_1 \\ \vdots \\ x_{m_1-2} \\ x_{m_1-1} \end{pmatrix} + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \frac{-y_{m_2-m_1-1}\beta_{m_1}\beta_{m_2+1}}{\beta_{m_1+1}}x_{m_2+1} \end{pmatrix} = \lambda(B_1)_1 \begin{pmatrix} x_1 \\ \vdots \\ x_{m_1-2} \\ x_{m_1-1} \end{pmatrix}, \tag{5.32}$$

and

$$\begin{aligned} &\beta_{m_1}x_{m_1-1} + \frac{-y_{m_2-m_1-1}\alpha_{m_1}\beta_{m_2+1}}{\beta_{m_1+1}}x_{m_2+1} + \beta_{m_1+1}x_{m_1+1} \\ &= \lambda b_{m_1}\frac{-y_{m_2-m_1-1}\beta_{m_2+1}}{\beta_{m_1+1}}x_{m_2+1}. \end{aligned} \tag{5.33}$$

Since $det(A_1^0) = 0$ and A_1^0 is unreduced tridiagonal, $det((A_1^0)^1) \neq 0$. Therefore

$$(A_1^0)^1 v = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \text{ and } (A_1^0)^1 w = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}$$

have solutions. Write $v = (v_1, ..., v_{m_2-m_1-1})^T$ and $w = (w_1, ..., w_{m_2-m_1-1})^T$. $(A_1^0)^1$ is symmetric, so is its inverse. So $w_1 = v_{m_2-m_1-1}$. From (5.28),

$$\begin{aligned} x_{m_1+2} &= -v_1\beta_{m_1+2}x_{m_1+1} - w_1\beta_{m_2+1}x_{m_2+1} \\ x_{m_2} &= -w_1\beta_{m_1+2}x_{m_1+1} - w_{m_2-m_1-1}\beta_{m_2+1}x_{m_2+1}. \end{aligned} \tag{5.34}$$

It follows from (5.31),

$$(A_1^0)^1 \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_{m_2-m_1-1} \end{pmatrix} = - \begin{pmatrix} \beta_{m_1+2} \\ 0 \\ \vdots \\ 0 \end{pmatrix},$$

thus,

$$y_{m_2-m_1-1} = -\beta_{m_1+2}v_{m_2-m_1-1} = -\beta_{m_1+2}w_1. \tag{5.35}$$

Substituting (5.34) into (5.29), we have

$$w_1\beta_{m_1+2}\beta_{m_2+1}x_{m_1+1} + (\alpha_{m_2+1} - w_{m_2-m_1-1}\beta^2_{m_2+1})x_{m_2+1} + \beta_{m_2+2}x_{m_2+2} = \lambda b_{m_2+1}x_{m_2+1}. \tag{5.36}$$

Similarly, substituting (5.35) into (5.32) and (5.33), yields,

$$\beta_{m_1}x_{m_1-1} + \frac{w_1\alpha_{m_1}\beta_{m_1+2}\beta_{m_2+1}}{\beta_{m_1+1}}x_{m_2+1} + \beta_{m_1+1}x_{m_1+1} = \lambda b_{m_1}\frac{w_1\beta_{m_1+2}\beta_{m_2+1}}{\beta_{m_1+1}}x_{m_2+1}, \tag{5.37}$$

and

$$(A_1)_1\begin{pmatrix} x_1 \\ \vdots \\ x_{m_1-2} \\ x_{m_1-1} \end{pmatrix} + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \frac{w_1\beta_{m_1}\beta_{m_1+2}\beta_{m_2+1}}{\beta_{m_1+1}}x_{m_2+1} \end{pmatrix} = \lambda(B_1)_1\begin{pmatrix} x_1 \\ \vdots \\ x_{m_1-2} \\ x_{m_1-1} \end{pmatrix}. \tag{5.38}$$

We may solve for x_{m_1+1} in (5.36) and substitute it into (5.37), to obtain

$$\frac{w_1\beta_{m_1}\beta_{m_1+2}\beta_{m_2+1}}{\beta_{m_1+1}}x_{m_1-1} + (\alpha_{m_2+1} - w_{m_2-m_1-1}\beta^2_{m_2+1} + \alpha_{m_1}(\frac{w_1\beta_{m_1+2}\beta_{m_2+1}}{\beta_{m_1+1}})^2)x_{m_2+1}$$

$$+\beta_{m_2+2}x_{m_2+2} = \lambda b x_{m_2+1} \tag{5.39}$$

where

$$b = b_{m_2+1} + b_{m_1}(\frac{w_1\beta_{m_1+2}\beta_{m_2+1}}{\beta_{m_1+1}})^2 > 0.$$

Let S and T be $(n - m_2 + m_1 - 1) \times (n - m_2 + m_1 - 1)$ matrices of forms

$$S = \left(\begin{array}{cccccc} 0 & & & & & \\ & \ddots & & & & \\ & & 0 & s & & \\ & & s & t & \beta_{m_2+2} & \\ & & & \beta_{m_2+2} & 0 & \\ & & & & & \ddots \\ & & & & & & 0 \end{array}\right) \begin{array}{l} \\ \\ \ldots(m_1-1)^{th} row \\ \ldots(m_1)^{th}\ \ row \\ \ldots(m_1+1)^{th} row \\ \\ \\ \end{array},$$

and

$$T = \begin{pmatrix} O & & \\ & b & \\ & & O \end{pmatrix} \begin{matrix} \\ \\ \end{matrix} \begin{matrix} \\ ...(m_1)^{th}row \\ \end{matrix},$$

where $s = \frac{w_1\beta_{m_1}\beta_{m_1+2}\beta_{m_2+1}}{\beta_{m_1+1}}$, $t = \alpha_{m_2+1} - w_{m_2-m_1-1}\beta^2_{m_2+1} + \alpha_{m_1}(\frac{w_1\beta_{m_1+2}\beta_{m_2+1}}{\beta_{m_1+1}})^2$.

From (5.32), (5.38) and (5.39), we have

$$\left(\begin{pmatrix} (A_1)_1 & & \\ & 0 & \\ & & (A_2)^1 \end{pmatrix} + S\right)\begin{pmatrix} x_1 \\ \vdots \\ x_{m_1-1} \\ x_{m_2+1} \\ \vdots \\ x_n \end{pmatrix} = \lambda\left(\begin{pmatrix} (B_1)_1 & & \\ & 0 & \\ & & (B_2)^1 \end{pmatrix} + T\right)\begin{pmatrix} x_1 \\ \vdots \\ x_{m_1-1} \\ x_{m_2+1} \\ \vdots \\ x_n \end{pmatrix}. \tag{5.40}$$

Obviously, the matrix on the left hand side in (5.40) is unreduced tridiagonal and the matrix on the right hand side is positive definite and diagonal. The eigenvalues of (5.40) are the same eigenvalues of the pencil (A, B).

Furthermore, after solving (5.40), we can calculate x_{m_1+1} from (5.36), x_{m_1} from (5.27) and $x_{m_1+2}, ..., x_{m_2}$ from (5.28). Hence, the eigenvector of the pencil (A, B) corresponding to the eigenvalue λ can be easily formed from the eigenvector of (5.40) corresponding to the same eigenvalue λ.

Q.E.D

Theorem 5.2 *Assume* $det(A_i^0) = 0$, *for* $i = 1, 2, ..., r$.

(i) If A and B are of the same form as in (5.12), then the pencil (A, B) *has the same eigenvalues as the eigenproblem:*

$$\left(\begin{pmatrix} (A_1)_1 & & & & \\ & 0 & & & \\ & & (A_2)^1_1 & & \\ & & & \ddots & \\ & & & & (A_r)^1 \end{pmatrix} + S\right) y = \lambda\left(\begin{pmatrix} (B_1)_1 & & & & \\ & 0 & & & \\ & & (B_2)^1_1 & & \\ & & & \ddots & \\ & & & & (B_r)^1 \end{pmatrix} + T\right) y \tag{5.41}$$

where S is a symmetric matrix, and the matrix on the left hand side is an unreduced symmetric tridiagonal, T is a diagonal matrix and the matrix on the right hand side

is positive definite and diagonal.

(ii) If A and B are of the same form as in (5.13), then the pencil (A, B) has the same eigenvalues as the following eigenproblem

$$\left(\begin{pmatrix} (A_1)_1 & & & & \\ & 0 & & & \\ & & (A_2)_1^1 & & \\ & & & \ddots & \\ & & & & (A_r)_1^1 \end{pmatrix} + \tilde{S}\right) y = \lambda \left(\begin{pmatrix} (B_1)_1 & & & & \\ & 0 & & & \\ & & (B_2)_1^1 & & \\ & & & \ddots & \\ & & & & (B_r)_1^1 \end{pmatrix} + \tilde{T}\right) y \tag{5.42}$$

where $\tilde{S}$ is a symmetric matrix, and the matrix on the left hand side in (5.42) is unreduced symmetric tridiagonal, $\tilde{T}$ is a diagonal matrix and the matrix on the right hand side is positive definite and diagonal.

Proof: By the same arguments used in Lemma 5.3 and Lemma 5.4, the proof can be easily achieved by mathematics induction.

Q.E.D.

Theorem 5.3 *If A is an unreduced symmetric tridiagonal matrix and B is a positive semidefinite diagonal matrix, then the pencil (A, B) can be reduced to a positive definite pencil $(\tilde{A}, \tilde{B})$, where $\tilde{A}$ is still unreduced symmetric tridiagonal and $\tilde{B}$ is positive definite diagonal.*

Proof: The proof follows immediately from Theorem 5.1 and Theorem 5.2.

Q.E.D.

Let f_n denote the number of the eigenvalues of the pencil (A, B), then by Theorem 5.1 and Theorem 5.2, we may obtain the following result.

Theorem 5.4

$$f_n = rank(B) - \sum_{i=1}^{r}(1 - sign(|det(A_i^0)|)).$$

Since B is diagonal, $rank(B)$ is just the number of nonzero diagonal elements of B. If zero is an eigenvalue of A_i^0, $sign(|det(A_i^0)|) = 0$. Otherwise, $sign(|det(A_i^0)|) = 1$.

Since A_i^0 is unreduced, the Sturm's sequence [24, 31] can be used to check if zero is an eigenvalue of A_i^0, so, f_n can be easily computed without actually solving the generalized eigenvalue problem.

Chapter 6

Preliminary Analysis

Let A be an $n \times n$ real symmetric tridiagonal matrix and B be an $n \times n$ positive definite diagonal matrix of the form

$$A = \begin{pmatrix} \alpha_1 & \beta_2 & & & O \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{n-1} & \alpha_{n-1} & \beta_n \\ O & & & \beta_n & \alpha_n \end{pmatrix}, \qquad B = \begin{pmatrix} b_1 & & & O \\ & b_2 & & \\ & & \ddots & \\ O & & & b_n \end{pmatrix}, \tag{6.1}$$

where $b_i > 0$, and $\beta_i \neq 0$ for all i. Our algorithm employed the same strategy of 'Divide and Conquer' as we did in Part I. First of all, the matrix A is divided into two blocks by letting one of the β_i's equal to zero. Namely, for the initial pencil (D, B) in (4.3), we let

$$D = \begin{pmatrix} D_1 & 0 \\ 0 & D_2 \end{pmatrix}, \tag{6.2}$$

where

$$D_1 = \begin{pmatrix} \alpha_1 & \beta_2 & & & O \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{k-1} & \alpha_{k-1} & \beta_k \\ O & & & \beta_k & \alpha_k \end{pmatrix}, D_2 = \begin{pmatrix} \alpha_{k+1} & \beta_{k+2} & & & O \\ \beta_{k+2} & \alpha_{k+2} & \beta_{k+3} & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{n-1} & \alpha_{n-1} & \beta_n \\ O & & & \beta_n & \alpha_n \end{pmatrix},$$

and then let

$$B = \begin{pmatrix} B_1 & 0 \\ 0 & B_2 \end{pmatrix}$$

with $dim(B_i) = dim(D_i), i = 1, 2$. We calculate the eigenvalues of the pencils (D_i, B_i), $i = 1, 2$, by using the most efficient algorithm available. Then our algorithm conquers the pencil (A, B) by the homotopy in (4.3).

Theorem 6.1 *The solution set of (4.3) consists of disjoint smooth curves and each curve joins one eigenpair of the pencil (D, B) and one eigenpair of the pencil (A, B).*

Proof: Differentiating $H(x, \lambda, t)$ with respect to (x, λ) in (4.3) yields,

$$H_{(x,\lambda)}(x, \lambda, t) = \begin{pmatrix} \lambda B - A(t) & Bx \\ x^T B & 0 \end{pmatrix}.$$

We claim that if (x, λ, t) satisfies $H(x, \lambda, t) = 0$, then $H_{(x,\lambda)}(x, \lambda, t)$ is nonsingular. To see this, we show that $H_{(x,\lambda)}(x, \lambda, t)y = 0$ has only trivial solution. Clearly $dim(ker(\lambda B - A(t))) = 1$ and $x^T Bx \neq 0$, since for each t, $\lambda B - A(t)$ is unreduced tridiagonal and B is positive definite. Suppose y satisfies $H_{(x,\lambda)}y = 0$, write $y = (y_1^T, y_2)^T$, where $y_1 \in \mathbf{R^n}$ and $y_2 \in \mathbf{R}$, then

$$(\lambda B - A(t))y_1 + y_2 Bx = 0 \tag{6.3}$$

$$x^T B y_1 = 0. \tag{6.4}$$

Since $x^T Bx \neq 0$ and $x^T(\lambda B - A(t)) = 0$, (6.3) implies $y_2 = 0$. Hence,

$$(\lambda B - A(t))y_1 = 0,$$

or

$$B^{\frac{1}{2}}(\lambda I - B^{-\frac{1}{2}} A(t) B^{-\frac{1}{2}}) B^{\frac{1}{2}} y_1 = 0.$$

That is, $B^{\frac{1}{2}} y_1 \in ker(\lambda I - \tilde{A}(t))$ with $\tilde{A}(t) = B^{-\frac{1}{2}} A(t) B^{-\frac{1}{2}}$. It is easy to see that $\tilde{A}(t)$ is also unreduced since $B^{-\frac{1}{2}}$ is a diagonal matrix with positive diagonal elements.

Thus, $dim(ker(\lambda I - \tilde{A}(t)) = 1$. On the other hand,

$$\begin{aligned}(\lambda I - \tilde{A}(t))B^{\frac{1}{2}}x &= (\lambda I - B^{-\frac{1}{2}}A(t)B^{-\frac{1}{2}})B^{\frac{1}{2}}x \\ &= (\lambda B^{\frac{1}{2}} - B^{-\frac{1}{2}}A(t))x \\ &= B^{-\frac{1}{2}}(\lambda B - A(t))x = 0.\end{aligned}$$

Thus, $B^{\frac{1}{2}}x \in ker(\lambda I - \tilde{A}(t))$ and hence, $B^{\frac{1}{2}}x = cB^{\frac{1}{2}}y_1$ for certain nonzero constant c. From (6.4),

$$x^T B y_1 = (x^T B^{\frac{1}{2}})(B^{\frac{1}{2}}y_1) = c(B^{\frac{1}{2}}y_1)^T(B^{\frac{1}{2}}y_1) = 0.$$

Therefore, $y_1 = 0$, since $B^{\frac{1}{2}}$ is positive definite. Hence $y = 0$.

A repeated application of the implicit function theorem, the assertion of the theorem is achieved.

Q.E.D.

Let ξ_i and $\lambda_i(t)$ be eigenvalues of the pencils (D, B) and $(A(t), B)$, respectively. Then the following theorems follow immediately from the results we proved in Part I.

Theorem 6.2 *If eigenvalues of (D, B) are distinct then*
(i) Either $\lambda_i(t)$ is constant for all t in [0,1] or strictly monotonic.
(ii) $\dot{\lambda}_i(t)\bar{\lambda}_i(t) > 0$ for t small, if $\dot{\lambda}_i(t) \not\equiv 0$.

Theorem 6.3 *For any $t \in [0, 1]$,*

$$\begin{aligned}&\xi_{i-1} \leq \lambda_i(t) \leq \xi_{i+1}, i = 2, 3, ..., n-1, \\ &\lambda_1(t) \leq \xi_1 \\ &\lambda_n(t) \geq \xi_n.\end{aligned}$$

From Theorem 6.2 and Theorem 6.3, every homotopy curve must be one of those in Figure 6.1. Each homotopy curve is bounded by two consecutive dotted lines and no homotopy curve can cross any dotted line.

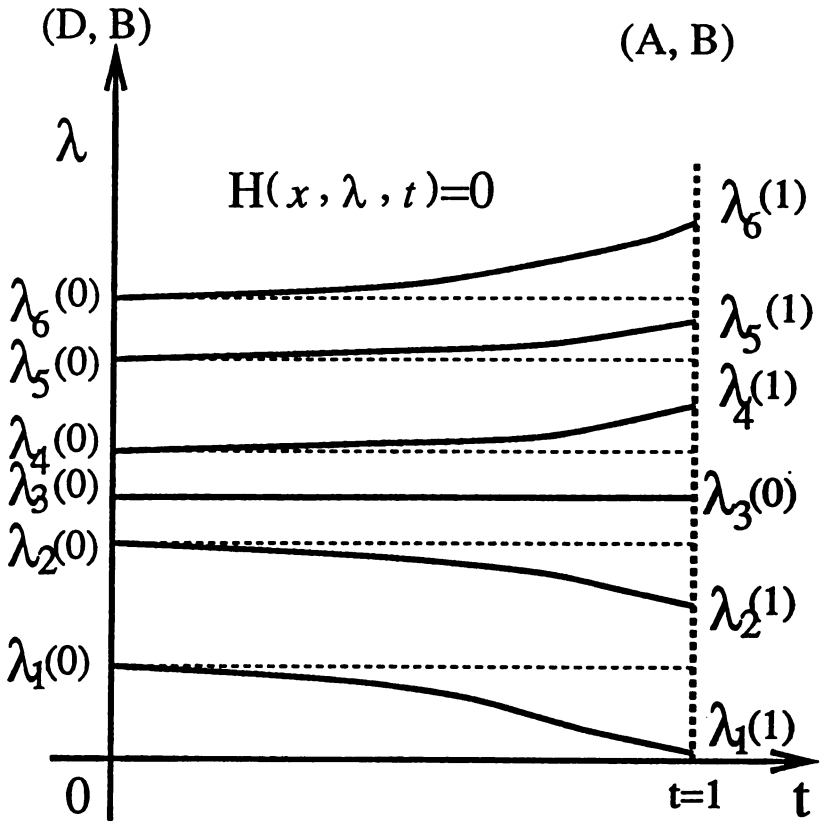

Figure 6.1: The generalized homotopy curves

Chapter 7

The Homotopy Algorithms for Generalized Eigenproblem

Our algorithm for finding all the eigenpairs of the pencil (A, B) with unreduced symmetric tridiagonal matrix A and positive semidefinite diagonal matrix B is based on the following steps:

(i) Reduction

(ii) Initiating at $t = 0$

(iii) Prediction

(iv) Correction

(v) Checking

(vi) Detection of a cluster

(vii) Step-size selection

(viii)Terminating at $t = 1$

(ix) Forming the eigenpairs of the pencil (A, B).

7.1 Reduction

If B is positive semidefinite, the pencil (A, B) will be reduced to a pencil $(\tilde{A}, \tilde{B})$ with $\tilde{B}$ positive definite by the reduction we described in Chapter 5.

7.2 Initiating at $t = 0$

As we mentioned in Part I, by making the initial matrix D in (6.2) close to A, then the eigenpairs of the pencil (D, B) should be excellent starting points as we mentioned in Part I. We intend to choose k for which β_{k+1} is as small as possible. To make the sizes of the blocks D_1 and D_2 roughly the same, we limit the choice of k in the range $n/2 - j \leq k \leq n/2 + j$, where j is roughly equal to $n/20$, and find the smallest β_{k+1} by local sorting.

When the initial matrix D is decided, we calculate the eigenvalues of the pencils (D_1, B_1) and (D_2, B_2) by using the most efficient method available. We only require the accuracy stay within one-half or even one-third of the working precision. With this strategy, considerable amount of computing time is saved.

7.3 Prediction

Assume that after i steps the approximate value $(\tilde{x}(t_i), \tilde{\lambda}(t_i))$ on the eigenpath $(x(t), \lambda(t))$ at t_i is known and the next step-size h is determined; that is, $t_{i+1} = t_i + h$. We want to find an approximate value $(\tilde{x}(t_{i+1}), \tilde{\lambda}(t_{i+1}))$ of $(x(t_{i+1}), \lambda(t_{i+1}))$ on the eigenpath at t_{i+1}. Notice that $(\tilde{x}(t_{i+1}), \tilde{\lambda}(t_{i+1}))$ is an approximate eigenpair of the pencil $(A(t_{i+1}), B)$. Since $H(x(t), \lambda(t), t) = 0$, we have

$$\begin{cases} A(t)x(t) = \lambda(t)Bx(t) \\ \\ x(t)^T Bx(t) = 1. \end{cases}$$

Differentiating both equations with respect to t, yields,

$$\begin{aligned} &(A - D)x(t) + A(t)\dot{x}(t) = \dot{\lambda}(t)Bx(t) + \lambda(t)B\dot{x}(t) \\ &x(t)^T B\dot{x}(t) = 0. \end{aligned} \tag{7.1}$$

For $t = t_i$, multiplying (7.1) on the left by $x^T(t_i)$, yields,

$$\dot{\lambda}(t_i) = x^T(t_i)(A - D)x(t_i) = 2\beta_{k+1}x_k(t_i)x_{k+1}(t_i) \tag{7.2}$$

where $x(t_i) = (x_1(t_i), ..., x_n(t_i))^T$. In view of (7.2), we use the Euler predictor to predict the eigenvalue at t_{i+1}, namely,

$$\lambda^0(t_{i+1}) = \lambda(t_i) + \dot{\lambda}(t_i)h.$$

It is easy to see that $\dot{\lambda}(0) = 0$ in (7.2). Consequently, $\lambda^0(t_1)$ always equals to $\lambda^0(0)$. To predict the eigenvector, we use the inverse iterations on $(A(t_{i+1}), B)$ with shift $\lambda^0(t_{i+1})$. That is, we solve

$$(A(t_{i+1}) - \lambda^0(t_{i+1})B)y^0(t_{i+1}) = x(t_i)$$

and let

$$x^0(t_{i+1}) = \frac{By^0(t_{i+1})}{||By^0(t_{i+1})||}.$$

At $t_i = 0$, since we skip the calculations of eigenvectors of the pencil (D, B), $x(0)$ is not available. We choose a random vector for $x(0)$.

7.4 Correction

When B is positive definite, the generalized Rayleigh quotient $\rho(u) = (u^TAu)/u^TBu$ enjoys the following properties [24]:

(i) Stationarity.

Since

$$grad(\rho(u)) \equiv \nabla(\rho(u)) = \frac{2(Au - \rho(u)Bu)^T}{u^TBu},$$

thus ρ is stationary at the eigenvectors of the pencil (A, B).

(ii) Minimum residual.

$$||(A - \sigma B)u||^2_{B^{-1}} \geq ||Au||^2_{B^{-1}} - |\rho(u)|^2||Bu||^2_{B^{-1}},$$

where $||x||^2_{B^{-1}} = x^TB^{-1}x$, with equality holds if and only if $\sigma = \rho(u)$.

(iii) Monotonicity.

$$||(A - \rho_{k+1}B)x_{k+1}||_{B^{-1}} \leq ||(A - \rho_kB)x_k||_{B^{-1}} \qquad \text{for all } k.$$

(iv) Cubic convergence.

Therefore, we use the generalized RQI as a corrector, starting with $x^0(t_{i+1})$. To be more precise, at $(x^{j-1}(t_{i+1}), \lambda^{j-1}(t_{i+1}))$ $(j \geq 1)$ let

$$\lambda^j(t_{i+1}) = x^{j-1}(t_{i+1})^T A(t_{i+1}) x^{j-1}(t_{i+1})$$

then solve

$$(A(t_{i+1}) - \lambda^j(t_{i+1})B)y^j(t_{i+1}) = x^{j-1}(t_{i+1})$$

and let

$$x^j(t_{i+1}) = \frac{By^j(t_{i+1})}{\|By^j(t_{i+1})\|}.$$

We repeat the above process to within half of the working precision if single precision is used and one-third of the working precision if double precision is used when $t_{i+1} < 1$, since precision in determining the curve itself is only of secondary interest. We polish $(x^j(t_{i+1}), \lambda^j(t_{i+1}))$ at the end of the path $(t_{i+1} = 1)$ by iterating the Rayleigh quotient to machine precision. The stop point $(x^j(t_{i+1}), \lambda^j(t_{i+1}))$ of RQI will be taken as an approximate eigenpair $(\tilde{x}(t_{i+1}), \tilde{\lambda}(t_{i+1}))$ of the pencil $(A(t_{i+1}), B)$.

The cubic convergence rate of RQI makes the corrector highly efficient.

7.5 Checking

For

$$A = \begin{pmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{n-1} & \alpha_{n-1} & \beta_n \\ & & & \beta_n & \alpha_n \end{pmatrix}, \quad \text{and} \quad B = \begin{pmatrix} b_1 & & & \\ & b_2 & & \\ & & \ddots & \\ & & & b_n \end{pmatrix},$$

with nonzero $\beta_i's$ and $b_i > 0$, the polynomials defined by

$$p_0(\lambda) = 1$$
$$p_1(\lambda) = \alpha_1 - \lambda b_1$$
$$p_r(\lambda) = (\alpha_r - \lambda b_r)p_{r-1}(\lambda) - \beta_r^2 p_{r-2}(\lambda)$$
$$r = 2, 3, ..., n$$

form a generalized Sturm sequence. Thus, the number of the eigenvalues of (A, B) strictly greater than λ is equal to the number of the sign changes of the Sturm sequence, with the convention that if $p_r(\lambda) = 0$, then $p_r(\lambda)$ is taken to have the opposite sign of $p_{r-1}(\lambda)$.

When $(\tilde{x}(t_{i+1}), \tilde{\lambda}(t_{i+1}))$ is taken as an approximate eigenpair of the pencil $(A(t_{i+1}), B)$, the generalized Sturm sequence at $\tilde{\lambda}(t_{i+1}) + \epsilon$ is computed to check that, if we are trying to follow the curve corresponding to j^{th} largest eigenvalue, we are still on that curve. Here, ϵ is chosen as half of the working precision if single precision is used and one-third of the working precision if double precision is used. If the check fails, we reduce the step size to $h/2$ and repeat the whole process once again beginning with the eigenvalue prediction in Section 7.3.

7.6 Detection of a cluster

At $t_i = 0$, when all the eigenvalues of the pencil (D, B)

$$\lambda_1(0) < \lambda_2(0) < ... < \lambda_n(0)$$

are available, we let $\delta = max(10^{-5}, 10^{-2}(\lambda_n(0) - \lambda_1(0))/n)$ if double precision is used (or $\delta = max(10^{-3}, 10^{-2}(\lambda_n(0) - \lambda_1(0))/n)$ if single precision is used). Set λ_i and λ_j in the same group if $|\lambda_i(0) - \lambda_j(0)| < \delta$. If the number of the eigenvalues in any group is bigger than 1, then a cluster is detected. At $t_i \neq 0$, or 1, when $(\tilde{x}(t_i), \tilde{\lambda}(t_i))$ is taken as an approximate eigenpair of the pencil $(A(t_i), B)$, after the checking step in Section 7.5, we compute the Sturm sequences at $\tilde{\lambda}(t_i) \pm \delta$ to find the number of eigenvalues of $(A(t_i), B)$ in the interval $(\tilde{\lambda}(t_i) - \delta, \tilde{\lambda}(t_i) + \delta)$. When this number is bigger than 1, a cluster of eigenvalues of $(A(t_i), B)$ is detected.

In those cases, the corresponding eigenvectors are ill-conditioned and such ill-condition can cause the inefficiency of the algorithm. We simply give up following the eigenpath and the corresponding eigenpair of the pencil (A, B) will be calculated at the end of the algorithm (see Section 7.8).

7.7 Step size selection

In the first attempt, we always choose the step size $h = 1 - t_i$ at $t_i < 1$. If after the prediction and the correction steps the checking step fails, we reduce the step size to $h/2$ as mentioned in Section 7.5. Since the initial pencil (D, B) is chosen to be so close to (A, B), from our experiences, most of the eigenpairs of the pencil (A, B) can be reached in one step, i.e., $h = 1$.

Very small step size can also cause the inefficiency of the algorithm. Therefore, we impose a minimum γ on the step size h. If $h < \gamma$, we simply give up following the eigenpath and the corresponding eigenpair of A will be calculated at the end of the algorithm (see Section 7.8). We usually choose $\gamma \approx 0.25$.

7.8 Terminating at $t = 1$

At $t = 1$, when an approximate eigenvalue $\tilde{\lambda}(1)$ is reached, we compute the Sturm sequence at $\tilde{\lambda}(1) + \epsilon$ with $\epsilon =$ machine precision to ensure the correct order. If the checking fails, we have jumped into a wrong eigenpath. More precisely, suppose we are following the i^{th} eigenpair, the checking algorithm detects that we have reached the j^{th} eigenpair instead. In this situation, we will save the j^{th} eigenpair before the step size is cut. By saving the j^{th} eigenpair, the computation of following the j^{th} eigenpair is no longer needed.

As mentioned in Section 7.6 and 7.7, we may give up following some eigenpaths to avoid adapting a step size that is too small or the situation when a cluster is encountered. Without any extra computation, we know exactly which eigenpairs are lost at $t = 1$. In order to find these eigenpairs, we first use the bisection to find the eigenvalues up to the half working precision and then use the inverse iteration and the RQI. If there is a cluster, then we do bisection to find the eigenvalues up to the machine precision, then use the inverse iteration to find the corresponding eigenvectors. In this case, to guarantee the orthogonality, we orthonormalize the eigenvectors belonging to the same cluster while we are using the inverse iteration.

7.9 Forming the eigenpairs of (A, B)

If B is positive semidefinite, the pencil (A, B) was reduced to a pencil $(\tilde{A}, \tilde{B})$ with $\tilde{B}$ positive definite. From Theorem 5.1 and Theorem 5.2, the eigenvalues of the pencil $(\tilde{A}, \tilde{B})$ are the same eigenvalues of the pencil (A, B). Although the eigenvectors are different, the eigenvectors of the pencil (A, B) can be easily formed from those of the pencil $(\tilde{A}, \tilde{B})$ with few computations. The formulas are given in Chapter 5.

Chapter 8

Numerical Results of the Homotopy Method for Symmetric Generalized Eigenproblem

We shall show the computational results comparing the homotopy continuation method GHOMO with QZ method. The computations were done on a Sun SPARC station 1.

Our testing examples consist of the following types of pencils (A, B):

Type 1. A is an unreduced symmetric tridiagonal matrix with both diagonal and off-diagonal elements being uniformly distributed random numbers between 0 and 1. B is a diagonal matrix with the first $n/2$ diagonal elements being uniformly distributed random numbers between 0 and 1, and the last $n/2$ being zeros.

Type 2. A is an unreduced symmetric tridiagonal matrix with both diagonal and off-diagonal elements being uniformly distributed random numbers between 0 and 1. B is a diagonal matrix with the first $3n/10$ and the last $3n/10$ diagonal elements being uniformly distributed random numbers between 0 and 1, and the rest being zeros.

Type 3. A is Toeplitz matrix $[1, 2, 1]$. B is a diagonal matrix with the first $n/2$ diagonal elements being 1, and the rest being zeros.

Type 4. A is Toeplitz matrix $[1, 2, 1]$. B is a diagonal matrix with the first $3n/10$

and the last $3n/10$ diagonal elements being 1, and the rest being zeros.

Type 5. A is an unreduced symmetric tridiagonal matrix with both diagonal and off-diagonal elements being uniformly distributed random numbers between 0 and 1. B is a diagonal matrix with all diagonal elements being random numbers between 0 and 1.

Table 8.1 shows the comparison in terms of speed with the QZ method. Table 8.2 shows the accuracy and orthogonality of the homotopy method. The homotopy method appears to be strongly competitive and leads in speed by a considerable margin in comparison with the QZ method in all the cases.

Matrix Type	Order N	Execution Time (second)		
		GHOMO	QZ	Ratio (QZ/GHOMO)
Matrix Type 1	50	0.23	6.49	28.21
	100	0.75	48.20	64.26
	200	2.85	328.19	115.15
	400	12.39	2546.06	205.49
Matrix Type 2	50	0.28	6.23	22.25
	100	1.14	45.67	40.06
	200	4.86	321.86	66.22
	400	15.24	1802.32	118.26
Matrix Type 3	50	0.31	6.15	19.83
	100	1.60	46.96	29.35
	200	6.46	352.67	54.59
	400	23.68	2796.63	118.10
Matrix Type 4	50	0.52	5.82	10.15
	100	2.18	44.20	20.27
	200	8.80	336.50	38.23
	400	34.84	2694.60	77.34
Matrix Type 5	50	0.72	11.17	15.51
	100	2.81	74.66	26.56
	200	12.74	488.53	38.34
	400	47.70	3866.14	81.05

Table 8.1: Execution Time (second) of computed eigenpairs of generalized eigenproblems.

Matrix	Order N	$\max_i \|\|Ax_i - \lambda_i Bx_i\|\|_2/\lambda_{max}$	$\max_{i,j} \|(X^T BX - I)_{i,j}\|/\lambda_{max}$
	50	3.9107905787880D-16	2.8617639677154D-16
Matrix	100	3.4241406339558D-16	2.6417938095675D-15
Type	200	4.3779590340543D-16	2.0037668814815D-15
1	400	5.2265929394497D-16	5.1552956553653D-15
	50	3.4923828496565D-16	1.3678005594614D-15
Matrix	100	4.1134132219334D-16	1.0126081452260D-15
Type	200	5.1246002221589D-16	1.8957582132953D-13
2	400	4.9856607322645D-16	1.0934604011632D-14
	50	1.1454950034247D-16	3.4233801826002D-16
Matrix	100	1.6564408035527D-16	1.1470895221836D-15
Type	200	2.1624971847800D-16	3.6096263986950D-15
3	400	2.4734721400459D-16	2.1752300627827D-14
	50	1.0336086782520D-16	2.4870346559132D-16
Matrix	100	2.0867616866469D-16	2.3841690810960D-15
Type	200	2.1954327917052D-16	6.4657627842233D-15
4	400	2.9860202732850D-16	2.6556085482256D-14
	50	2.1611472844832D-16	2.2100520796971D-15
Matrix	100	4.3084863486435D-16	1.7434653812219D-15
Type	200	4.6005546307787D-16	5.2256957128428D-15
5	400	6.0472727089284D-16	2.1235908790871D-14

Table 8.2: The residual and orthonormality of computed eigenvectors of generalized eigenproblems.

Bibliography

[1] R. Barlow and D. Evans, *A parallel organization of the bisection algorithm,* The Computer Journal, 22(1977), pp. 267-269.

[2] H. Bernstine and M. Goldstine, *Parallel implementation of bisection for the calculation of eigenvalues of tridiagonal symmetric matrices,* Computing, 37(1986), pp. 85-91.

[3] A. Bounse-Gerstner, *An algorithm for the symmetric generalized eigenvalue problem,* Linear Alg. Appl., 58(1984), pp 43-68.

[4] H. Bowdler, R. Martin, and J. Wilkinson, *The QR and QL algorithms for symmetric matrices,* Numer. Math, 11(1968), pp. 227-240.

[5] M. T. Chu, *A simple Application of the Homotopy Method to Symmetric Eigenvalue Problems,* Linear Algebra and Appl., Vol. 59 (1984), 85-90.

[6] M. T. Chu, *A Note on the Homotopy Method for Linear Algebraic Eigenvalue Problems,* Linear Algebra and Appl., Vol. 105, (1988), 225-236.

[7] M. T. Chu, T. Y. Li and T Sauer, *Homotopy Method for General λ-Matrix Problems,* SIAM J. Matrix Anal. and Appl., Vol., 9, No. 4, 528-536, (1988).

[8] J. J. M. Cuppen, *A divide and conquer method for the symmetric tridiagonal eigenproblem,* Numer. Math., 36(1981), pp.177-1951.

[9] J. J. Dongarra and D. C. Sorensen, *A fully parallel algorithm for symmetric eigenvalue problems,* SIAM J. Sci. Stat. Comput. Vol 8. (1987), pp. 139-154.

[10] G. Fix and R. Heiberger, *An algorithm for the ill-conditioned generalized eigenvalue problem,* SIAM, J. Numer. Anal., 9(1972), pp 78-88.

[11] F. Rellich, *Perturbation theory of eigenvalue problems,* Gordon and Breach Science Publishers, New York, NY, 1969.

[12] G. Golub and C. V. Loan, *Matrix computations,* The Johns Hopkins Press, Baltimore, MD, 1983.

[13] R. O. Hill, *Interlacing of eigenvalues of symmetric, tridiagonal matrices,* A Preprint.

[14] I. Ilse and E. Jessup, *Solving the symmetric tridiagonal eigenvalue problem on the Hypercube,* SIAM J. Sci. Stat. Comput. vol 11. (1990), pp. 203-229.

[15] E. Jessup, *Parallel Solution of the Symmetric Tridiagonal Eigenproblem,* Ph.D. Thesis, (1989) Yale University.

[16] K. Li and T. Y. Li, *An Algorithm For Symmetric Tridiagonal Eigen-problems — Divide and Conquer with Homotopy Continuation,* A Preprint.

[17] T. Y. Li, *The homotopy method for solving real symmetric generalized eigenvalue problem,* A Preprint.

[18] T. Y. Li and N. H. Rhee, *Homotopy algorithm for symmetric eigenvalue problems,* Numer.Math. 55, (1989), pp.256-280.

[19] T. Y. Li and T. Sauer, *Homotopy method for generalized eigenvalue problems,* Lin. Alg. Appl. 91(1987), pp. 65-74.

[20] T. Y. Li, T. Sauer and J. Yorke, *Numerical Solution of a Class of Deficient Polynomial Systems,* SIAM J. Numer. Anal. 24, 435-451 (1987).

[21] T. Y. Li, Z. Zeng, and L. Cong, *Solving Eigenvalue Problems of Real Nonsymmetric Matrices with Real Homotopy,* A Preprint.

[22] T. Y. Li, H. Zhang and X. H. Sun, *Parallel homotopy algorithm for symmetric tridiagonal eigenvalue problem,* to appear: SIAM J. Sci. Stat. Comput.

[23] S. S. Lo, B. Philippe and A. Sameh, *A multiprocessor algorithm for symmetric tridiagonal eigenvalue problem,* SIAM J. Sci. Stat. Comput. Vol 8. (1987), pp. 155-165.

[24] B. N. Parlett, *The symmetric eigenvalue problem,* Englewood Cliffs, N.J., Prentice-Hall (1980).

[25] A. Sameh and D. Kuck, *A parallel QR algorithm for symmetric tridiagonal matrices,* IEEE Trans. Computers, C-26(1977), pp. 147-53.

[26] G. Peters and J. H. Wilkinson, $Ax = \lambda Bx$ *and the generalized eigenproblem,* SIAM J. Numer. Anal., 7(1970), pp 479-492.

[27] Smith, B. T., et.al., *Matrix eigensystem routines-EISPACK Guide.* 2nd edition,New york, Springer-Verlag (1976).

[28] D. C. Sorensen and Peter Tang, *On the Orthogonality of Eigenvectors Computed by Divide-and-Conquer Techniques,* A Preprint.

[29] G. Stewart, *Introduction to matrix computations,* Academic Press, Now York, 1973.

[30] J. Stoer and R. Bulirsch, *Introduction to numerical analysis,* Springer-Verlag, 1980.

[31] J. H. Wilkinson, *The algebraic eigenvalue problem,* New York, Oxford University Press (1965).

[32] J. H. Wilkinson and C. Reinsch, *Handbook for automatic computation: Linear Algebra,* Springer-Verlag, 1971.