Internat. J. Math. & Math. Sci.
VOL. 17 NO. 4 (1994) 741-752

A FULLY PARALLEL METHOD FOR TRIDIAGONAL EIGENVALUE PROBLEM

KUIYUAN LI

Department of Mathematics and Statistics
University of West Florida
Pensacola, FL 32514

(Received July 23, 1992)

ABSTRACT. In this paper, a fully parallel method for finding all eigenvalues of a real matrix pencil (A, B) is given, where A and B are real symmetric tridiagonal and B is positive definite. The method is based on the homotopy continuation coupled with the strategy 'Divide-Conquer' and Laguerre iterations. The numerical results obtained from implementation of this method on both single and multiprocessor computers are presented. It appears that our method is strongly competitive with other methods. The natural parallelism of our algorithm makes it an excellent candidate for a variety of advanced architectures.

KEY WORDS AND PHRASES. Eigenvalues, Eigenvalue curves, Multiprocessors, Homotopy method.
1992 AMS SUBJECT CLASSIFICATION CODE. 65F15.

1. INTRODUCTION.

When B is a well-conditioned positive definite matrix, real symmetric generalized eigenvalue problem

$$Ax = \lambda Bx \tag{1}$$

can be reduced to the form

$$L^{-1}AL^{-T}(L^Tx) = \lambda(L^Tx) \tag{2}$$

where A and B are real $n \times n$ symmetric matrices and $B = LL^T$. There are many very efficient algorithms for (2), for instant, the QR algorithm [8], the D&C algorithm [3], the bisection algorithm [5] and the homotopy algorithm [6]. When A and B are both tridiagonal the above technique is unattractive because $L^{-1}AL^{-T}$ is, in general, a full matrix.

In this paper, we shall present a parallel homotopy method for finding all the eigenvalues or all eigenpairs of a matrix pencil (A,B), where A and B are both real symmetric tridiagonal and B is positive definite. Assume in (1),

$$A = \begin{pmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{n-1} & \alpha_{n-1} & \beta_n \\ & & & \beta_n & \alpha_n \end{pmatrix} \text{ and } B = \begin{pmatrix} \delta_1 & \gamma_2 & & & \\ \gamma_2 & \delta_2 & \gamma_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \gamma_{n-1} & \delta_{n-1} & \gamma_n \\ & & & \gamma_n & \delta_n \end{pmatrix} \tag{3}$$

If $\beta_i = \gamma_i = 0$ for some i, then (1) can clearly be decomposed into two subproblems and we can solve them independently. Hence, we will assume all β_i and γ_i are not both equal to zero.

Let

$$C = \begin{pmatrix} A_1 & 0 \\ 0 & A_2 \end{pmatrix}$$

where

$$A_1 = \begin{pmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{k-1} & \alpha_{k-1} & \beta_k \\ & & & \beta_k & \alpha_k \end{pmatrix}, \quad A_2 = \begin{pmatrix} \alpha_{k+1} & \beta_{k+2} & & & \\ \beta_{k+2} & \alpha_{k+2} & \beta_{k+3} & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{n-1} & \alpha_{n-1} & \beta_n \\ & & & \beta_n & \alpha_n \end{pmatrix},$$

$$D = \begin{pmatrix} B_1 & 0 \\ 0 & B_2 \end{pmatrix}, \tag{5}$$

where

$$B_1 = \begin{pmatrix} \delta_1 & \gamma_2 & & & \\ \gamma_2 & \delta_2 & \gamma_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \gamma_{k-1} & \delta_{k-1} & \gamma_k \\ & & & \gamma_k & \delta_k \end{pmatrix}, \quad B_2 = \begin{pmatrix} \delta_{k+1} & \gamma_{k+2} & & & \\ \gamma_{k+2} & \delta_{k+2} & \gamma_{k+3} & & \\ & \ddots & \ddots & \ddots & \\ & & \gamma_{n-1} & \delta_{n-1} & \gamma_n \\ & & & \gamma_n & \delta_n \end{pmatrix}.$$

Consider the homotopy $H: \boldsymbol{R} \times [0\ ,\ 1] \to \boldsymbol{R}$, defined by

$$\begin{aligned} H(\lambda, t) &= det((1-t)(C - \lambda D) + t(A - \lambda B)) \\ &= det(A(t) - \lambda B(t)), \end{aligned}$$

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

In section 2, we shall show that the solution set of $H(\lambda, t) = 0$ in (6) consists of continuously differentiable curves $\lambda(t)$, each joins an eigenvalue of (C,D) to one of (A,B). We call each of these curves a *homotopy curve* or an *eigencurve*. We shall also show that each eigenvalue curve is monotonic in t. And if m, the multiplicity of an eigencurve $\lambda(t)$ is greater than one in any subinterval of $[0,1]$, then it must be a constant curve. In the consequence, it is an eigenvalue of (A,B) of multiplicity of m or $m+1$. We shall give the details of our algorithm in section 3 and some numerical results will be presented in section 4.

2. PRELIMINARY ANALYSIS

Proposition 2.1. *Let $H(\lambda, t)$ be defined as in (6), then the solution set of $H(\lambda, t) = 0$ consists of real, continuously differentiable eigencurves.*

PROOF. First of all, we show that $\lambda(t)$, a solution of $H(\lambda(t), t) = 0$ is real for any t in $[0,1]$. Since $H(\lambda(t), t) = det(A(t) - \lambda(t)B(t))$, we only need to show that $B(t)$ is positive definite for all t in $[0,1]$. Let $\lambda_1(t)$ be the smallest eigenvalue

of $B(t)$, then by Cauchy's interlace theorem [8], $\lambda_1(0) > \lambda_1(t)$ for all t in $(0,1]$. By Proposition 2.1 in [6], $\lambda_1(t)$ is strictly monotonic in t. Therefore, $\lambda_1(t)$ is strictly monotone decreasing in t. Hence $\lambda_1(t) > \lambda_1(1) > 0$ for all t in $[0,1)$ since $B(1) = B$ is positive definite. Hence $B(t)$ is positive definite for all t in $[0,1]$.

Now we show that $\lambda(t)$ is continuously differentiable. Clearly, $H(\lambda,t)$ can be written as:

$$H(\lambda(t),t) = c_n(t)\lambda^n(t) + c_{n-1}(t)\lambda^{n-1}(t) + \cdots + c_1(t)\lambda(t) + c_0(t),$$

where $c_i(t)$'s are polynomials in t. Let $\lambda(t)$ be a solution of $H(\lambda(t),t) = 0$ and t_0 any point in $(0,1)$. By Puiseux's theorem [10],

$$\lambda(t_0+\epsilon) = \lambda(t_0) + b_1\epsilon^{\frac{1}{h}} + b_2\epsilon^{\frac{2}{h}} + \cdots$$

converges for sufficiently small ϵ. Let b_m denote the first nonzero coefficient, then

$$b_m = \lim_{\epsilon \to 0+} \frac{\lambda(t_0+\epsilon) - \lambda(t_0)}{\epsilon^{\frac{m}{h}}}$$

is real since $\lambda(t_0+\epsilon)$, $\lambda(t_0)$ and ϵ are all real. On the other hand,

$$(-1)^{\frac{m}{h}} b_m = \lim_{\epsilon \to 0-} \frac{\lambda(t_0+\epsilon) - \lambda(t_0)}{(-\epsilon)^{\frac{m}{h}}}$$

is also real. Hence, $(-1)^{\frac{m}{h}}$ is a real number. Therefore, m must be a multiple of h. We can continue the same argument to show that only integral powers of ϵ can have nonzero coefficients. Therefore, $\lambda(t_0+\epsilon) = \sum_{i=0}^{\infty} b_i\epsilon^i$. Hence, $\lambda(t)$ is continuously differentiable.

Q.E.D.

PROPOSITION 2.2. *Any eigencurve $\lambda(t)$ of (A(t), B(t)) is either straight line or strictly monotonic in t.*

PROOF: Since

$$A(t) - \lambda B(t) = \left(\begin{array}{c|c|c} A_1 - \lambda(t)B_1 & t\alpha & \\ \hline t\alpha & & \\ \hline & & A_2 - \lambda(t)B_2 \end{array}\right),$$

where $\alpha = \beta_{k+1} - \lambda(t)\gamma_{k+1}$, there exist polynomials $f_1(\lambda)$ and $f_2(\lambda)$ such that $H(\lambda,t) = f_1(\lambda) - t^2 f_2(\lambda)$. If there exists a λ_0 such that $f_2(\lambda_0) = 0$, then $H(\lambda_0,t) = 0$ implies $f_1(\lambda_0) = 0$. Hence $H(\lambda_0,t) = 0$ for all t in $[0,1]$. Therefore, $\lambda(t) = \lambda_0$ for all t in $[0,1]$, i.e., $\lambda(t)$ is a straight line.

If for any λ, $f_2(\lambda) \neq 0$, then $H(\lambda_0,t) = 0$ has at most one solution in $[0,1]$. Therefore, $\lambda(t)$ must be strictly monotonic in t. Otherwise, for certain λ_0, $H(\lambda_0,t) = 0$ will have more than one solution in $[0,1]$ (See Figure 1).

Q.E.D.

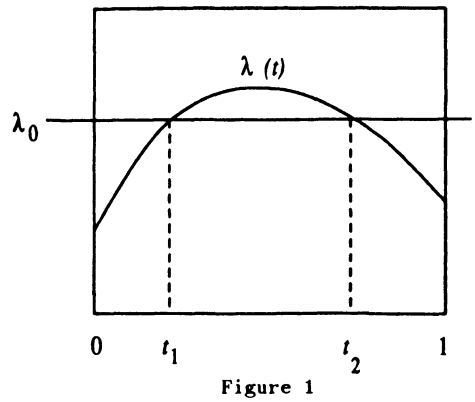

Figure 1

Let $m(\lambda(t))$ denote the multiplicity of $\lambda(t)$. Proposition 2.2 implies that if $m(\lambda(t)) = 1$ and there are t_1 and t_2 in [0,1] such that $\lambda(t_1) = \lambda(t_2)$, where $t_1 \neq t_2$, then $\lambda(t)$ must be constant.

PROPOSITION 2.3. *Let $\lambda(t)$ be an eigencurve. If [a,b] is a subinterval of [0,1] and $m(\lambda(t)) = m > 1$, for any t in [a,b], then $\lambda(t) \equiv \lambda(0)$ and $\lambda(0)$ is an eigenvalue of matrix pencil (A, B) of multiplicity of m or $m + 1$.*

PROOF. Claim 1: $\lambda(t) \equiv \lambda_0$ for some constant λ_0, for t in [a,b].

Let $t_1, t_2, \cdots, t_n$ be n distinct points in [a,b]. Since *rank* $(A(t) - \lambda(t)B(t)) = n - m < n - 1$ for any t in [a,b] and $A(t) - \lambda(t)B(t)$ is symmetric tridiagonal, at least one of the off-diagonal elements of $A(t) - \lambda(t)B(t)$, say $\beta_s - \lambda(t)\gamma_s$, is equal to zero at two points, say t_i and t_j. Otherwise, *rank* $(A(t) - \lambda(t)B(t)) \geq n - 1$. Hence, $\beta_s - \lambda(t_i)\gamma_s = 0$ and $\beta_s - \lambda(t_j)\gamma_s = 0$. Since β_s and γ_s are not both equal to zero, $\lambda(t_i) = \lambda(t_j)$. Hence, $\lambda(t) \equiv \lambda_0$ for some constant λ_0, for any t in [a,b] by Proposition 2.2.

Claim 2: $rank(A(t) - \lambda_0 B(t)) = n - m$ for all t in (0,1] except at possible one point in (0,1] $rank(A(t) - \lambda_0 B(t)) = n - m - 1$.

If $\beta_{k+1} - \lambda_0 \gamma_{k+1} = 0$, then clearly, $rank(A(t) - \lambda_0 B(t)) = n - m$ for all t in (0,1]. If $\beta_{k+1} - \lambda_0 \gamma_{k+1} \neq 0$, since some off diagonal elements of $A(t) - \lambda_0 B(t)$ are equal to zero, $A(t) - \lambda_0 B(t)$ can be rewritten as:

$$A(t) - \lambda_0 B(t) = \begin{pmatrix} A_1 - \lambda_0 B_1 & & \\ & A_2(t) - \lambda_0 B_2(t) & \\ & & A_3 - \lambda_0 B_3 \end{pmatrix}$$

where $A_2(t) - \lambda_0 B_2(t)$ is unreduced symmetric tridiagonal, i.e., all its off-diagonal elements are not equal to zero.

Hence,

$$rank(A(t) - \lambda_0 B(t)) = rank(A_1 - \lambda_0 B_1) + rank(A_2(t) - \lambda_0 B_2(t)) + rank(A_3 - \lambda_0 B_3)$$

Clearly, $rank(A_1 - \lambda_0 B_1) + rank(A_3 - \lambda_0 B_3)$ is constant. Assume $rank(A_2(t) - \lambda_0 B_2(t)) = m_0$ for some integer m_0 for all t in [a,b]. Since $(A_2(t) - \lambda_0 B_2(t))$ is a unreduced symmetric tridiagonal matrix, its eigencurves are smooth, disjoint and monotonic in t by Proposition 2.1 in [6]. Therefore, if $\mu(t)$ is an eigencurve of

$(A_2(t)-\lambda_0 B_2(t))$, i.e., $det((A_2(t)-\lambda_0 B_2(t)) - \mu(t)) = 0$, then either $\mu(t) \equiv 0$ or there is at most one t such that $\mu(t) = 0$. Hence, if zero is an eigenvalue of $(A_2(t) - \lambda_0 B_2(t))$ for some t in [a,b], then $\mu(t) = 0$ is an eigenvalue for any t in [0,1] since $rank(A_2(t)-\lambda_0 B_2(t)) = m_0$ for all t in [a,b]. Therefore, $rank(A_2(t)-\lambda_0 B_2(t)) = m_0$ for all t in (0,1]. If for any t in (0,1], zero is not an eigenvalue of $(A_2(t)-\lambda_0 B_2(t))$, then $rank(A_2(t)-\lambda_0 B_2(t)) = m_0$. Hence, $rank(A_2(t)-\lambda_0 B_2(t)) = n-m$ for all t in (0,1] in these two cases. If for any t in [a,b], zero is not an eigenvalue of$(A_2(t)-\lambda_0 B_2(t))$, then clearly, there is at most one t in (0,1] such that $rank(A_2(t)-\lambda_0 B_2(t)) = m_0 - 1$.

Hence, $\lambda(t)$ is constant for all t in [0,1] and $\lambda(0)$ is an eigenvalue of (A,B) of multiplicity of m or $m+1$.

Q.E.D.

From Proposition 2.3, we will not have any those bifurcation curves in Figure 2.

If $\lambda(t)$ is not constant, then its multiplicity must be 1.

Let $\lambda_1(t)$, $\lambda_2(t)$, ... , $\lambda_n(t)$ be n eigencurves of $(A(t),B(t))$, where $\lambda_1(0) \leq \lambda_2(0) \leq$, ... , $\leq \lambda_n(0)$.

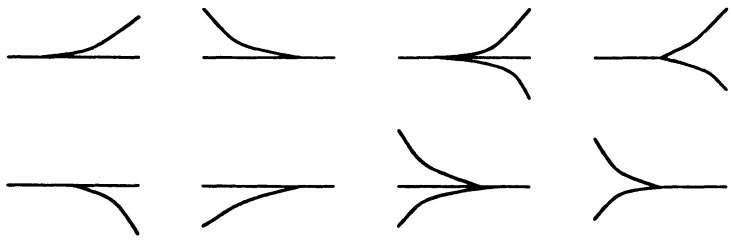

Figure 2

PROPOSITION 2.4. *Let $\lambda_i(t)$ and $\lambda_{i+1}(t)$ be two nonconstant eigencurves of (A(t), B(t)). If $\lambda_i(t_0) < \lambda_{i+1}(t_0)$ for some t_0 in (0,1], then $\lambda_i(t) < \lambda_{i+1}(t)$ for all t in (0,1].*

PROOF. From Proposition 2.2 and 2.3, both $\lambda_i(t)$ and $\lambda_{i+1}(t)$ must be strictly monotonic in t. Assume $\lambda_i(t)$ and $\lambda_{i+1}(t)$ are both strictly monotone increasing. If there exists a t_1 in (0,1] such that $\lambda_i(t_1)$ and $\lambda_{i+1}(t_1)$, then $\lambda_i(t_1)$ and $\lambda_{i+1}(0)$ since $\lambda_{i+1}(t)$ is strictly monotonic in t. Therefore, $t_2 > 0$ is a solution of $H(\lambda_{i+1}(0),t) = 0$ (See Figure 3). By Proposition 2.2, $\lambda_{i+1}(t)$ is a constant. Contradiction.

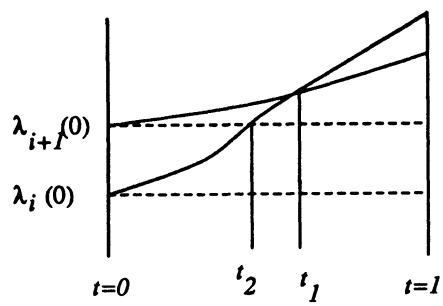

Figure 3

Similarly, if $\lambda_i(t)$ and $\lambda_{i+1}(t)$ are both strictly monotone decreasing, then the conclusion holds.

Assume $\lambda_i(t)$ is monotone increasing and $\lambda_{i+1}(t)$ monotone decreasing, then $\lambda_i(0) < \lambda_{i+1}(0)$. If there is a $t_1 > 0$ such that $\lambda_i(t_1)$ and $\lambda_{i+1}(t_1)$, then there exists a $t_2 > t_1$ such that $\lambda_{i+1}(t_1) < \lambda_i(t_2) < \lambda_{i+1}(0)$. Since $\lambda_{i+1}(t)$ is continuous and $\lambda_{i+1}(t_1) < \lambda_{i+1}(t) < \lambda_{i+1}(0)$ for any t in $(0, t_1)$, there is a t_3 in $(0, t_1)$ such that $\lambda_{i+1}(t_3) = \lambda(t_2)$ (See Figure 4), then $H(\lambda_i(t_2), t) = 0$ has two solutions. Therefore, either $\lambda_i(t)$ or $\lambda_{i+1}(t)$ must be constant. Contradiction.

Clearly, if $\lambda_{i+1}(t)$ is increasing and $\lambda_i(t)$ decreasing, then the conclusion holds since $\lambda_i(0) < \lambda_{i+1}(0)$.

Q.E.D.

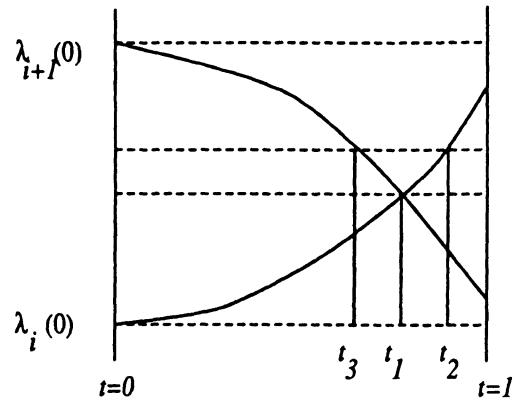

Figure 4

From Proposition 2.4, if $\lambda_i(t)$ and $\lambda_{i+1}(t)$ are both constants or both are not constants, then they must be disjoint. However, if one of them is constant, they may cross.

Example 1.

$$A = \begin{pmatrix} 4 & 1 & 0 \\ 1 & 1 & 4 \\ 0 & 4 & 1 \end{pmatrix}, \qquad B = \begin{pmatrix} 4 & 1 & 0 \\ 1 & 3 & 0 \\ 0 & 0 & 3 \end{pmatrix}.$$

Let

$$A(t) = \begin{pmatrix} 4 & 1 & 0 \\ 1 & 1 & 4t \\ 0 & 4t & 1 \end{pmatrix}, \qquad B(t) = \begin{pmatrix} 4 & 1 & 0 \\ 1 & 3 & 0 \\ 0 & 0 & 3 \end{pmatrix},$$

then $(A(t), B(t))$ has three eigencurves:

$$\lambda_1(t) = \frac{20 - \sqrt{4 + 8448t^2}}{66},$$

$$\lambda_2(t) = \frac{20 + \sqrt{4 + 8448t^2}}{66},$$

and

$$\lambda_3 = 1.$$

From these propositions, we know that all eigenvalues of the initial pencil not only close to the eigenvalues of pencil (A, B), but also separate them. These results provide us very important information in designing our code.
(See Figure 5).

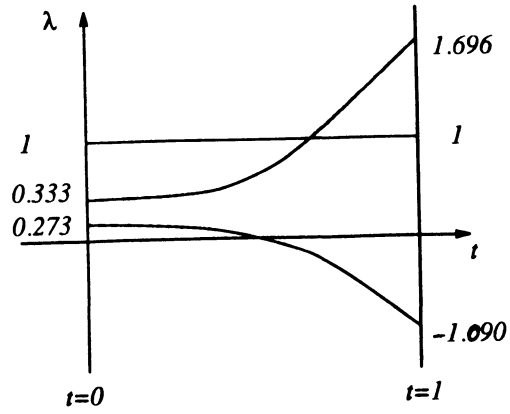

Figure 5

3. ALGORITHM

Our algorithm is based on following steps.

(*i*) Form initial matrix pencil.

For a given matrix pencils (A,B), let $k=[n/2]$, then form an initial matrix pencil as in (4) and (5) in Section 1. If k is greater than 2, repeat above procedure on those submatrices, until the dimension of each submatrices is less than or equal to 2. Now, (C,D) is the initial matrix pencil, where $C = diag(A_{s,1}, A_{s,2}, \ldots, A_{s,p})$ and $D = diag(B_{s,1}, B_{s,2}, \ldots, B_{s,p})$, $p = 2^s$ for some positive integer s. We have a tree (See Figure 6).

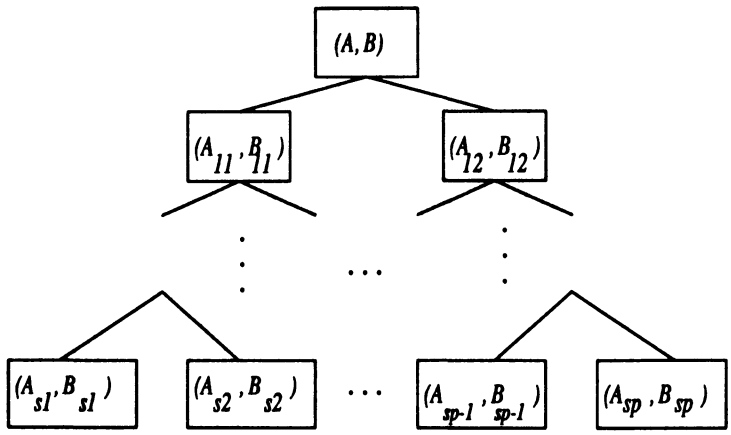

Figure 6

(*ii*) Compute all eigenvalues of (C,D).

Compute the eigenvalues of each submatrix pencil $(A_{s,i}, B_{s,i})$, $i=1, 2, \ldots,p$. Since A_{si} and B_{si} are at most 2×2 matrices, the eigenvalues of pencil (A_{si},B_{si}) can be easily obtained.

(*iii*) Conquer.

After all the eigenvalues of each submatrix pencil (A_{si},B_{si}) $i = 1,2,\ldots,p$ are available, we then compute all the eigenvalues of each submatrix pencil (A_{si}, B_{si}), $i=1,2,\ldots,p/2$ by simple step homotopy method with Laguerre iterations.

It is sufficient to assume that all the eigenvalues of matrix pencils (A_{11}, B_{11}) and (A_{12}, B_{12}) are available and we want to compute all eigenvalues of matrix pencil (A,B).

Let

$$C = \begin{pmatrix} A_{11} \\ A_{12} \end{pmatrix} \quad \text{and} \quad D = \begin{pmatrix} B_{11} \\ B_{12} \end{pmatrix} .$$

Construct a homotopy as in (6):

$$\begin{aligned} H(\lambda, t) &= det((1-t)(C-\lambda D) + t(A-\lambda B)) \\ &= det(A(t)-\lambda B(t)), \end{aligned}$$

where $A(t) = (1-t)C + tA$ and $B(t) = (1-t)D + tB$.

Let $\lambda_1(t)$, $\lambda_2(t)$, ... , $\lambda_n(t)$ be n eigencurves of $H(\lambda,t)$, where $\lambda_1(0) \le \lambda_2(0) \le ,..., \le \lambda_n(0)$ are the eigenvalues of matrix pencil (C,D). We conquer (C,D) to (A,B) with Laguerre iteration [12], i.e., use the eigenvalues of (C,D) as starting points of Laguerre iterations to compute all eigenvalues of (A,B).

(a) Some knowledge of the Laguerre iteration.

Let $f(\lambda)$ be a function having n real zeros $\lambda_1 < \lambda_2 <, ..., < \lambda_n$ and x_0 lie between λ_i and λ_{i+1}, then Laguerre iteration gives the two sequences

$$x_{m+1} = x_m - \frac{nf(x_m)}{f'(x_m) \pm \sqrt{(n-1)^2(f'(x_m))^2 - n(n-1)f(x_m)f''(x_m)}} . \tag{7}$$

One of which converges to λ_i, another to λ_{i+1} monotonicly. In the neighborhood of a simple zero at λ_i, (7) converges cubicly to λ_i. On the other hand, in the neighborhood of a zero λ_i of multiplicity r,

$$x_{m+1} = x_m - \frac{nf(x_m)}{f'(x_m) + sign(f'(x_m))\sqrt{\frac{(n-r)}{r}((f'(x_m))^2 - nf(x_m)f''(x_m))}}$$

also converges to λ_i cubicly.

(b) Computing $f(\lambda)$, $f'(\lambda)$ and $f''(\lambda)$.

Let $f(\lambda) = det(A-\lambda B)$, then all the zeros of $f(\lambda)$ are real. Since

$$A-\lambda B = \begin{pmatrix} \alpha_1-\lambda\delta_1 & \beta_2-\lambda\gamma_2 & & & \\ \beta_2-\lambda\gamma_2 & \alpha_2-\lambda\delta_2 & \beta_3-\lambda\gamma_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{n-1}-\lambda\gamma_{n-1} & \alpha_{n-1}-\lambda\delta_{n-1} & \beta_n-\lambda\gamma_n \\ & & & \beta_n-\lambda\gamma_n & \alpha_n-\lambda\delta_n \end{pmatrix} ,$$

we have the following three term recursion.

$p_0(\lambda) = 1$

$p_1(\lambda) = (\alpha_1 - \lambda\delta_1)$

$p_m(\lambda) = (\alpha_m - \lambda\delta_m)p_{m-1}(\lambda) - (\beta_m - \lambda\gamma_m)^2 p_{m-2}(\lambda)$, $m = 2, \dots, n$

$f(\lambda) = p_n(\lambda)$.

By simple computation, we have following:

$p_0'(\lambda) = 0$

$p_1'(\lambda) = -\delta_1$

$$p'_m(\lambda) = (\alpha_m - \lambda\delta_m)p'_{m-1}(\lambda) - \delta_m p_{m-1}(\lambda) - (\beta_m - \lambda\gamma_m)^2 p'_{m-2}(\lambda) + 2(\beta_m - \lambda\gamma_m)\gamma_m p_{m-2}(\lambda)$$
$$m = 2,3,\ldots,n$$
$$f'(\lambda) = p'_n(\lambda)$$

and

$$p''_0(\lambda) = 0$$
$$p''_1(\lambda) = 0$$
$$p''_m(\lambda) = (\alpha_m - \lambda\delta_m)p''_{m-1}(\lambda) - (\beta_m - \lambda\gamma_m)^2 p''_{m-2}(\lambda) - 2\delta_m p'_{m-1}(\lambda) + 4(\beta_m - \lambda\gamma_m)\gamma_m p'_{m-2}(\lambda) - 2\gamma_m^2 p_{m-2}(\lambda)$$
$$m = 2,3,\ldots,n$$
$$f''(\lambda) = p''_n(\lambda)$$

By computing $p_m(\lambda), m = 0,1,2,\ldots,n$, we know exactly how many eigenvalues of (A,B) are less than λ. Since x_m converges monotonicly, after one Laguerre iteration, we know exactly x_m converges to which eigenvalue of (A,B) without any extra computations.

(c) The overflow and the underflow control.

Since $f(\lambda) = det(A-\lambda B)$, $f(\lambda)$ could be very small or large. Since $f(\lambda)$ can be written as

$$f(\lambda) = \Pi_{i=1}^{n} a_i(\lambda - \lambda_i),$$
$$f'(\lambda = \sum_{j=1}^{n} \Pi_{i\neq j}^{n} a_i(\lambda - \lambda_i)$$

and

$$f''(\lambda) = \sum_{k=1}^{n} \sum_{j=1}^{n} \Pi_{i\neq j,k}^{n} a_i(\lambda - \lambda_i).$$

We can see that we can find β such that $f(\lambda) = \alpha 10^{\beta}$, $f'(\lambda) = \gamma 10^{\beta}$ and $f''(\lambda) = \delta 10^{\beta}$, where α, γ and δ have magnitude between the machine minimum and maximum numbers. Hence, when we compute $p_m(\lambda)$, $p'_m(\lambda)$ and $p''_m(\lambda)$, if $p_m(\lambda)$ is too large (or too small), then we multiply $p_m(\lambda)$, $p'_m(\lambda)$ and $p''_m(\lambda)$ by 10^{β} for some integer β so that $p_m(\lambda)10^{\beta}$ is not too large (or too small). In this way, we can avoid overflow as well as underflow. Finally, we will get $\tilde{p}_n$, $\tilde{p}'_n(\lambda)$ and $\tilde{p}''_n(\lambda)$ and all of them have magnitude between machine minimum and maximum numbers and $f(\lambda) = 10^s \tilde{p}_n$, $f'(\lambda) = 10^s \tilde{p}'_n$ and $f''(\lambda) = 10^s \tilde{p}''_n$, for some integer s. Since

$$x_{m+1} = x_m - \frac{nf(x_m)}{f'(x_m) \pm \sqrt{(n-1)^2(f'(x_m))^2 - n(n-1)f(x_m)f''(x_m)}}$$

$$= x_m - \frac{n\tilde{p}_n(x_m)}{\tilde{p}'_n(x_m) \pm \sqrt{(n-1)^2(\tilde{p}'_n(x_m))^2 - n(n-1)\tilde{p}(x_m)\tilde{p}''_n(x_m)}},$$

overflow and underflow can be avoided.

Now we give details of computing eigenvalues of (A,B).

Step 1. If $i \neq 1$, go to step 2. Use $\lambda_1(0)$ as a starting point to do Laguerre iteration to locate μ_1, the smallest eigenvalue of (A,B). By computing $f(\lambda_1(0))$, we know exactly if $\{x_m\}$, the sequence generated by Laguerre iterations, will converge to μ_1. If it does not, then use $\lambda_1(0) - \alpha$ as a starting point, where $\alpha = 2|\beta_{k+1}/\gamma_{k+1}|$ if $\gamma_{k+1} \neq 0$ and $\alpha = 2|\beta_{k+1}|$ if $\gamma_{k+1} = 0$. If $\lambda_1(0) < \mu_1$, $\{x_m\}$ will converge to μ_1.

Step 2. If $\lambda_i(0)$ is not a simple eigenvalue of (C,D), then go to step 3. If $|\lambda_i(0) - \mu_{i-1}| > \epsilon$, where ϵ = tolerance* $\max\{|\lambda_n(0)|, |\lambda_1(0)|\}$, use $\lambda_i(0)$ as a starting point. Then, $\{x_m\}$ will either converge to μ_i or μ_{i+1}, since the starting point is either in the neighborhood of μ_i or μ_{i+1} according to the propositions 2.2 and

2.4 in section 2. If it converges to μ_i, then set $i=i+1$, go to step 5. If it converges to μ_{i+1}, then use $(\mu_{i+1}+\mu_{i-1})/2$ as a starting point to do Laguerre iteration and the new sequence $\{x_m\}$ will converge to μ_i. Set $i=i+2$, go to step 5.

Step 3. If the multiplicity r of $\lambda_i(0)$ is equal to 2, go to step 4. by the propositions in last section, $\lambda_i(0)$ is an eigenvalue of (A,B). The generalized Sturm sequences at $\lambda_i(0)\pm\epsilon$ are computed on (A,B) to check the multiplicity of $\lambda_i(0)$. We may have following cases according to the proposition 2.3.

(a). $\mu_i=\mu_{i+1}=\ldots=\mu_{i+r-1}=\lambda_i(0)$. Then set $i=i+r$, go to step 5.

(b). $\mu_{i+1}=\mu_{i+2}=\ldots=\mu_{i+r-1}=\lambda_i(0)$. Then use $(\lambda_i(0)+\mu_{i-1})/2$ as a starting point to do Laguerre iterations and $\{x_m\}$ will converge to μ_i. Then set $i=i+r$, go to step 5.

(c). $\mu_i=\mu_{i+1}=\ldots=\mu_{i+r-2}=\lambda_i(0)$. Then use $(\lambda_i(0)+\lambda_{i+r}(0))/2$ as a starting point and $\{x_m\}$ will converge to μ_{i+r-1}. Then set $i=i+r$, go to step 5.

(d). $\mu_{i+1}=\mu_{i+2}=\ldots=\mu_{i+r-2}=\lambda_i(0)$. Then use $(\mu_{i+1}+\mu_{i-1})/2$ as a starting point to do Laguerre iterations and $\{x_m\}$ will converge to μ_i. Then use $(\lambda_i(0)+\lambda_{i+r}(0))/2$ as a starting point and $\{x_m\}$ will converge to μ_{i+r-1}. Then set $i=i+r$, go to step 5.

Step 4. The generalized Sturm sequence at $\lambda_i(0)\pm\epsilon$ are computed on (A,B) to check if $\lambda_i(0)$ is an eigenvalue of (A,B). If it is not a eigenvalue of (A,B), then Two sequences $\{x_m\}$ and $\{x'_m\}$ will be generated by Laguerre iterations with $\lambda_i(0)$ as a starting point. One of them converges to μ_i and another to μ_{i+1}. Then set $i=i+2$, go to step 5.

If $\lambda_i(0)$ is an eigenvalue of (A,B), then we may have following cases according to the proposition 2.2.

(a). $\mu_i=\mu_{i+1}=\lambda_i(0)$.

Then set $i=i+2$, go to step 5.

(b). $\mu_{i+1}=\lambda_i(0)$.

Then use $(\mu_{i+1}+\mu_{i-1})/2$ as a starting point to do Laguerre iterations. $\{x_m\}$, generated by Laguerre iteration, will converge to μ_i. Then set $i=i+2$, go to step 5.

(c). $\mu_i=\lambda_i(0)$ and $\mu_{i+1}\neq\lambda_i(0)$.

Use $(\lambda_{i+2}(0)+\mu_i)/2$ as a starting point to do Laguerre iterations. $\{x_m\}$, generated by Laguerre iteration, will converge to μ_{i+1}. Then set $i=i+2$, go to step 5.

Step 5. If $i>n$, go to (*vi*). Otherwise, go to step 2.

(*vi*) Compute eigenvectors.

If the eigenvectors are required, compute them by inverse iteration.

4. NUMERICAL RESULTS

In this section, we present our numerical results. Our homotopy 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.

EXPERIMENT 1. We implemented our algorithm on 50 pencils (A,B) on each different dimension, where A is symmetric tridiagonal random matrix with both diagonal and off-diagonal elements being uniformly distributed random numbers between 0 and 1. B is a symmetric tridiagonal matrix with off-diagonal elements, γ_i, being uniformly distributed random numbers between 0 and 1, and its diagonal elements $\delta_i=2max(\gamma_i,\gamma_{i+1})$.

The computations were done on a Sun SPARC station 1.

Table 1 shows the results of our algorithm HRST and the algorithm RSG in EISPACK [4]. The algorithm RSG first reduces $Ax = \lambda Bx$ to $\widetilde{A}y = \lambda y$, then solves it. Since $\widetilde{A}$ is a full matrix, the RSG is unattractive for this problem as we mentioned in section 1. But the RSG is the only algorithm available for this problem in EISPACK.

Order	Execution Time all eigenvalues		Execution Time all eigenpairs		$\max_i \| Ax_i - \lambda_i Bx_i \|_2 / \lambda_{max}$		$\max_{i,j} \lvert (X^T X - I)_{i,j} \rvert$	
N	RGS	HRST	RGS	HRST	RGS	HRST	RGS	HRST
60	4.25	1.9	6.9	2.1	3.16D-14	8.32D-15	1.19D-14	4.91D-14
121	35.4	5.8	69.6	7.1	3.56D-14	1.75D-14	1.42D-14	1.63D-14
180	95.4	10.8	185.6	13.4	5.41D-14	2.83D-15	1.84D-14	8.02D-14
241	287.4	18.7	429.4	22.4	5.57D-14	7.10D-14	2.66D-14	5.73D-14

Table 1:Average execution time (second) of computed eigenvalues and eigenvectors.

Matrix	Order	Execution Time(sec.)		Ratio	$(\sum_{i=1}^{n} (\alpha(i) - \lambda(i)))/\lambda_{max}$	
Type	N	HRST	DSTEBZ	(DSTEBZ)/HRST)	HRST	DSTEBZ
[1,2,1]	65	0.76	2.42	3.18	- 1.2219D-15	- 5.3290D-15
	125	2.81	8.57	3.05	3.2201D-15	1.0658D-14
	255	11.35	34.78	3.06	- 8.6600D-15	- 2.1316D-14
	499	42.50	130.21	3.06	3.8858D-15	7.9936D-15
Wilkinson	65	0.73	1.52	2.08	- 2.6906D-14	1.1368D-13
	125	1.92	5.12	2.67	- 7.2474D-15	5.6843D-14
	255	6.70	19.34	2.89	6.6745D-16	2.8421D-14
	499	21.68	72.45	3.34	- 2.2760D-16	2.5871D-15

Table 2: The results of comparison of HRST with DSTEBZ.

EXPERIMENT 2. (a) A is the Toeplitz matrix [1, 2, 1], i.e., all diagonal elements, $\alpha(i)$, are 2 and off-diagonal elements are 1. $B = I$.

(b) A is the Wilkinson matrix, i.e., the matrix [1, $\alpha(i)$, 1], where $\alpha(i) = abs((n+1)/2 - i)$, $i = 1,2,\ldots,n$ with n odd. $B = I$.

Table 2 shows the computational results comparing our algorithm HRST with the bisection algorithm DSTEBZ in LAPACK [1] on these two problems. It appears that our algorithm leads in speed by a considerable margin in comparison with the DSTEBZ.

EXPERIMENT 3. Matrices A and B are obtained from piece wise linear finite element [11] discretization of the Sturm-Liouvill problem

$$-\frac{d}{dx}\left(p(x)\frac{du}{dx}\right) + q(x)u = \lambda u,$$

where $u = u(x)$, $0 < x < \pi$ and $u(0) = u'(\pi) = 0$ and $p(x) > 0$. Here, both A and B are symmetric tridiagonal and positive definite. We use $p(x) = 1$ and $q(x) = 6$.

The computations were executed on BUTTERFLY GP 1000, a shared memory multiprocessor machine.

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 cover p.

Order	$n = 500$					$n = 1000$					
Nodes	1	4	8	16	32	1	4	8	16	32	64
ExeTime	368.4	96.64	53.48	30.99	18.75	1584.	407.9	217.7	120.7	70.76	45.78
S_p	1.0	3.81	6.89	11.9	19.7	1.0	3.88	7.28	13.1	22.4	34.6
S_p/p	1.0	0.95	0.86	0.74	0.61	1.0	0.97	0.91	0.82	0.70	0.54

Table 3: Execution time(second), speed-up and efficiency of the algorithm HRST.

Table 3 shows the execution time and the speed-up S_p as well as the efficiency S_p/p of our algorithm. It appears that our algorithm is very efficient. The natural parallelism of our algorithm makes it an excellent candidate for multiprocessor machines.

ACKNOWLEDGEMENT: The research was supported in part by the Summer Research Grant from the University of West Florida. Author would like to acknowledge the valuable suggestions from discussions with Tien-Yien Li and Zhonggang Zeng.

REFERENCES

1. ANDERSON, E., et.al., *LAPACK User's Guide*, SIAM, Philadelphia, 1992.

2. CHU, M.T., LI, T.Y., and SAUER, T. *Homotopy Method for General λ-Matrix Problems*, SIAM J. Matrix Anal. and Appl., Vol., 9, No. 4, 528-536, (1988).

3. DONGARRA, J.J. and SORENSEN, D.C. *A fully parallel algorithm for symmetric eigenvalue problems*, SIAM J. Sci. Stat. Comput. Vol. 8 (1987) s139-s154.

4. GARBOW, B.S., et. al. *Matrix eigensystem routines-EISPACK Guide Extension*, New York, Springer-Verlag (1977).

5. ILSE, I. and JESSUP, E. *Solving the symmetric tridiagonal eigenvalue problem on the Hypercube*, SIAM J. Sci. Stat. Comput. Vol. 11 (1990), 203-229.

6. LI, K. and LI, T.Y. *An Algorithm for Symmetric Tridiagonal Eigen-problems – Divide and Conquer with Homotopy Continuation.* To appear on SIAM, J. Sci. Stat. Comput.

7. LI, T.Y. and SAUER, T. *Homotopy method for generalized eigenvalue problems*, Linear Alg. Appl. 91(1987), 65-74.

8. PARLETT, B.N. *The symmetric eigenvalue problem.* Prentice-Hall, Inc., Englenwood Cliffs, NJ (1980).

9. PETERS, G. and WILKINSON, J.H. *$Ax = \lambda Bx$ and the generalized eigenproblem*, SIAM J. Numer. Anal., 7(1970), 479-492.

10. RELLICH, F. *Perturbation theory of eigenvalue problems*, Gordon and Breach Science Publishers, New York, NY (1969).

11. STRANG, G. and FIX, G. *An analysis of the finite element method.* Prentice-Hall, Inc., Englenwood Cliffs, NJ (1973).

12. WILKINSON, J.H. *The algebraic eigenvalue problem*, New York, Oxford University Press (1965).