A ROOT-FINDING METHOD FOR EIGENPROBLEM

Kuiyuan Li*
Department of Mathematics and Statistics
University of West Florida
Pensacola, FL 32514, U.S.A.
E-mail: kli@uwf.edu

and

Tien-Yien Li†
Department of Mathematics
Michigan State University
East Lansing, MI 48824, U.S.A.
E-mail: li@math.msu.edu

الخلاصة :

نورد في هذه المقالة مثيلاً للطريقة العامة لايجاد كافة القيم المميزة. وتعتمد هذه الطريقة على أسلوب إيجاد الجذور والمُحدِّدة. تُوْصَفُ هذه الطريقة بأنها رتيبة التقارب، وهي تربيعية فَعَّالة وسهلة التطبيق.

ABSTRACT

In this paper we present a method, parallel in nature, for finding all eigenvalues of the generalized eigenvalue problem. Our method employs the determinant evaluation and a root-finding scheme. It is shown that the method converges monotonically and quadratically, and is reliable, efficient, and easy to implement in practice.

AMS(MOS) subject classification: 65F15.

* To whom correspondence should be addressed.

† This research was supported in part by the NSF under Grant DMS-9504953 and a Guggenheim Fellowship.

Kuiyuan Li and Tien-Yien Li

A ROOT-FINDING METHOD FOR EIGENPROBLEM

1. INTRODUCTION

In this paper we consider the generalized eigenvalue problem:

$$Ax = \lambda Mx, \tag{1}$$

where A and M are both real symmetric and one of them, say M, is positive definite, and both are banded with bandwidth k. We further assume $k \ll n$. The assumptions imply that the eigenvalues are real and an M-orthogonal basis of eigenvectors exists.

Eigenvalue problem with such a special structure arises in many applications, such as: numerical solutions of Sturm–Liouville and radial Schrödinger equations [1–3]; and the finite element approximation for free longitudinal vibrations problem of a non-uniform rod [4].

Consider the Sturm–Liouville problem,

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

If the finite difference approximation scheme is used, Equation (1) with A symmetric tridiagonal and M identity is obtained. On the other hand, if the piecewise linear finite element discretization [3] is used, we will achieve Equation (1) with both A and M symmetric tridiagonal and positive definite.

Consider the slightly more general problem [5],

$$Lu + \lambda r(x)u = 0, \quad u(a) = u(b) = 0,$$

where L is a differential operator, and $r(x) > 0$ and continuous on $[a, b]$. When a more general difference approximation,

$$L_h u_j + \lambda M_h u_j = 0, \quad j{=}1,2,\ldots,J,$$

is used, where $L_h u_j$ is an approximation to Lu and $M_h u_j$ is some other difference expression which is an approximation to $r(x_j)u(x_j)$, we will have Equation (1) with both A and M symmetric and banded, and M positive definite.

Let us consider equations of higher order than two, such as:

$$L_{2n}u - \lambda L_{2s}u = 0, \tag{2}$$

where $n > s$,

$$L_{2n}u = \sum_{k=0}^{n}(-1)^k \frac{d^k}{dx^k}\left(p_k(x)\frac{d^k u}{dx^k}\right),$$

and

$$L_{2s}u = \sum_{k=0}^{s}(-1)^k \frac{d^k}{dx^k}\left(q_k(x)\frac{d^k u}{dx^k}\right).$$

If the finite difference method is used to approximate the solution, we will have Equation (1) with both A and M symmetric, banded, and positive definite.

Similarly, use of the finite difference scheme to approximate the solution of the differential equation,

$$f(D)u = \lambda g(D)u,$$

where $f(D)$ and $g(D)$ are linear differential operators with g being of lower order than f, will result in Equation (1) with both A and M banded, symmetric, and positive definite.

Certain traditional approaches to solve (1) exist. The first approach is to reduce (1) to a standard eigenvalue problem:

$$L^{-1}AL^{-T}(L^Tx) = \lambda(L^Tx) \tag{3}$$

where $M = LL^T$ is the Cholesky factorization of M. Then eigenvalue problem (3) can be solved by many very efficient algorithms, such as the QR algorithm [6, 7], the bisection/multisection algorithms [8], and the homotopy algorithms [9–11]. However, this approach is less attractive because it can not take advantage of the banded form of A and M, and a full matrix $L^{-1}AL^{-T}$ is generated in the process. Furthermore, the accuracy of this method also depends on the conditioning of M since the inverse of L is explicitly required.

The second approach, the QZ method [12], disregards the symmetry and the banded structure of the matrices. After a direct phase requiring $O(n^3)$ operations, the algorithm enters an iterative phase requiring $O(n^3)$ operations to simultaneously reduce A and M to triangular-quasi-triangular form. While the cost is quite high, the algorithm is less sensitive to the conditioning of M.

In this paper we propose a method that computes all eigenvalues of (1) by finding zeroes of the polynomial equation:

$$p(\lambda) = \frac{1}{\alpha}det(A - \lambda M) \tag{4}$$

where $\alpha = (-1)^n det(M)$. Since M is positive definite, $\alpha \neq 0$.

Our method can be divided into two stages: eigenvalue isolation and eigenvalue extraction. The eigenvalue isolation can be done by bisections or multisections of order n_p, the number of processors.

For given μ_i, we compute a tridiagonal matrix T_i which is unitarily similar to $(A - \mu_i M)$ by using the Givens transformations.

The usual tridiagonalization process of a matrix obtained from the Householder transformations is less efficient for a banded matrix since the banded structure can not be maintained. It also involves an $O(n^3)$ computations instead of an expected complexity of $O(n^2k)$, where k denotes the bandwidth. Since $(A - \mu_i M)$ is indefinite in general, the usual LDL^T decomposition with diagonal matrix D may not exist. For banded matrices, the Givens rotations can successively eliminate the entries in the band. The sequence of rotations is generated when an entry is eliminated it may fill in a new nonzero entry lower down on the side of the band and this new entry must be eliminated by a subsequent rotation.

After $(A - \mu_i M)$ is reduced to a tridiagonal matrix T_i, the Sturm sequences are computed. As a result, the number of roots of $p(\lambda)$ in (4) which are smaller than μ_i and the number of roots in each subinterval are obtained. The process is recursively applied to every subinterval containing more than one root. This step results in a list of intervals containing single roots. When multiple roots are present, the isolation process continues until the required precision is reached. The eigenvalue isolation process has been discussed in details in [13].

The eigenvalue extraction step uses a root-finding method. In Section 2, we will show that our method, parallel in nature, converges quadratically, and is reliable and efficient.

In Section 3, we shall present our numerical results along with comparisons with other methods. It appears that our algorithm is strongly competitive in terms of speed and accuracy, 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 large degree parallel. It makes the parallel implementation of the method much simpler than the other methods. The numerical results suggest that our algorithm appears to be an excellent candidate for a variety of architectures.

2. EIGENVALUE EXTRACTION

Rewrite

$$p(\lambda) = \frac{1}{\alpha} det(A - \lambda M),$$

as

$$p(\lambda) = \lambda^n + a_1\lambda^{n-1} + \cdots + a_{n-1}\lambda + a_n. \tag{5}$$

with zeroes λ_1, λ_2, ..., λ_n.

After the eigenvalues are isolated, we assume that $\lambda_i \in (x_i^{(0)}, y_i^{(0)})$, for $i = 1, 2, ..., n$.

To find zeros of $p(x)$, we propose the following root-finding method:

$$x_i^{(k+1)} = x_i^{(k)} - \frac{p(x_i^{(k)})}{(x_i^{(k)}-x_1^{(k)})\cdots(x_i^{(k)}-x_{i-1}^{(k)})(x_i^{(k)}-y_{i+1}^{(k)})\cdots(x_i^{(k)}-y_n^{(k)})}, \quad i = 1, 2, ..., n \tag{6}$$

and

$$y_i^{(k+1)} = y_i^{(k)} - \frac{p(y_i^{(k)})}{(y_i^{(k)}-x_1^{(k)})\cdots(y_i^{(k)}-x_{i-1}^{(k)})(y_i^{(k)}-y_{i+1}^{(k)})\cdots(y_i^{(k)}-y_n^{(k)})}, \quad i = 1, 2, ..., n, \tag{7}$$

where the numbers $x_i^{(k)}$ and $y_i^{(k)}$, $i = 1, 2, \cdots, n$, are sets of approximations to n zeroes of $p(x)$. The numbers $x_i^{(k+1)}$ and $y_i^{(k+1)}$, $i = 1, 2, \cdots, n$, are sets of better approximations than $x_i^{(k)}$, and $y_i^{(k)}$, $i = 1, 2, \cdots, n$, respectively.

Clearly, (6) and (7) give us a fully parallel method for finding all zeros of $p(x)$, that is, $x_i^{(k+1)}$ and $y_i^{(k+1)}$, $i = 1, 2, \cdots, n$ can be computed independently.

Theorem 2.1 *Both $\{x_i^{(k)}\}_{k=0}^{\infty}$ and $\{y_i^{(k)}\}_{k=0}^{\infty}$ converge to λ_i strictly monotonically and quadratically.*

Proof. For any i,

$$\begin{aligned} x_i^{(k+1)} &= x_i^{(k)} - \frac{p(x_i^{(k)})}{(x_i^{(k)}-x_1^{(k)})...(x_i^{(k)}-x_{i-1}^{(k)})(x_i^{(k)}-y_{i+1}^{(k)})...(x_i^{(k)}-y_n^{(k)})} \\ &= x_i^{(k)} - \frac{(x_i^{(k)}-\lambda_1)(x_i^{(k)}-\lambda_2)...(x_i^{(k)}-\lambda_n)}{(x_i^{(k)}-x_1^{(k)})...(x_i^{(k)}-x_{i-1}^{(k)})(x_i^{(k)}-y_{i+1}^{(k)})...(x_i^{(k)}-y_n^{(k)})} \\ &= x_i^{(k)} - \frac{(x_i^{(k)}-\lambda_1)...(x_i^{(k)}-\lambda_{i-1})}{(x_i^{(k)}-x_1^{(k)})...(x_i^{(k)}-x_{i-1}^{(k)})} \frac{(x_i^{(k)}-\lambda_{i+1})...(x_i^{(k)}-\lambda_n)}{(x_i^{(k)}-y_{i+1}^{(k)})...(x_i^{(k)}-y_n^{(k)})} \cdot (x_i^{(k)} - \lambda_i). \end{aligned}$$

Since

$$\frac{(x_i^{(k)}-\lambda_1)...(x_i^{(k)}-\lambda_{i-1})}{(x_i^{(k)}-x_1^{(k)})...(x_i^{(k)}-x_{i-1}^{(k)})} > 0,$$

$$\frac{(x_i^{(k)}-\lambda_{i+1})...(x_i^{(k)}-\lambda_n)}{(x_i^{(k)}-y_{i+1}^{(k)})...(x_i^{(k)}-y_n^{(k)})} > 0,$$

and

$$(x_i^{(k)} - \lambda_i) < 0,$$

$$x_i^{(k+1)} > x_i^{(k)}.$$

Since

$$\frac{(x_i^{(k)}-\lambda_1)\ldots(x_i^{(k)}-\lambda_{i-1})}{(x_i^{(k)}-x_1^{(k)})\ldots(x_i^{(k)}-x_{i-1}^{(k)})} < 1,$$

$$\frac{(x_i^{(k)}-\lambda_{i+1})\ldots(x_i^{(k)}-\lambda_n)}{(x_i^{(k)}-y_{i+1}^{(k)})\ldots(x_i^{(k)}-y_n^{(k)})} < 1,$$

and

$$(x_i^{(k)} - \lambda_i) < 0,$$

therefore,

$$x_i^{(k+1)} < x_i^{(k)} - (x_i^{(k)} - \lambda_i) = \lambda_i.$$

Since $x_i^{(k)} < x_i^{(k+1)} < \lambda_i$ for $k = 0, 1, 2, \ldots$, $x_i^{(k)}$ converges.

Since

$$x_i^{(k+1)} = x_i^{(k)} - \frac{p(x_i^{(k)})}{(x_i^{(k)}-x_1^{(k)})\cdots(x_i^{(k)}-x_{i-1}^{(k)})(x_i^{(k)}-y_{i+1}^{(k)})\cdots(x_i^{(k)}-y_n^{(k)})},$$

$x_i^{(k)}$ must converge to λ_i.

Let

$$\varepsilon = \max_{1\le j\le n} \{\lambda_j - x_j^{(k)}, y_j^{(k)} - \lambda_j\}.$$

Then for any i,

$$\begin{aligned}
|x_i^{(k+1)} - \lambda_i| &= \left|(x_i^{(k)} - \lambda_i) - \frac{(x_i^{(k)}-\lambda_1)(x_i^{(k)}-\lambda_2)\ldots(x_i^{(k)}-\lambda_n)}{(x_i^{(k)}-x_1^{(k)})\ldots(x_i^{(k)}-x_{i-1}^{(k)})(x_i^{(k)}-y_{i+1}^{(k)})\ldots(x_i^{(k)}-y_n^{(k)})}\right| \\
&= \left|(x_i^{(k)} - \lambda_i) - \prod_{j=1}^{i-1} \frac{x_i^{(k)}-\lambda_j}{x_i^{(k)}-x_j^{(k)}} \prod_{j=i+1}^{n} \frac{x_i^{(k)}-\lambda_j}{x_i^{(k)}-y_j^{(k)}} \cdot (x_i^{(k)} - \lambda_i)\right| \\
&= \left|(x_i^{(k)} - \lambda_i) - \prod_{j=1}^{i-1} \left(1 + \frac{x_j^{(k)}-\lambda_j}{x_i^{(k)}-x_j^{(k)}}\right) \prod_{j=i+1}^{n} \left(1 + \frac{x_j^{(k)}-\lambda_j}{x_i^{(k)}-y_j^{(k)}}\right) \cdot (x_i^{(k)} - \lambda_i)\right| \\
&= |(x_i^{(k)} - \lambda_i)| \left|\sum_{j=1}^{i-1} \frac{x_j^{(k)}-\lambda_j}{x_i^{(k)}-x_j^{(k)}} + \sum_{j=i+1}^{n} \frac{x_j^{(k)}-\lambda_j}{x_i^{(k)}-y_j^{(k)}} + O(\varepsilon^2)\right| \\
&\le |x_i^{(k)} - \lambda_i| \left|\varepsilon \cdot \left(\sum_{j=1}^{i-1} \frac{1}{x_i^{(k)}-x_j^{(k)}} + \sum_{j=i+1}^{n} \frac{1}{y_j^{(k)}-x_i^{(k)}}\right) + O(\varepsilon^2)\right|
\end{aligned}$$

$$
\begin{aligned}
&\leq \varepsilon\left(\varepsilon\left(\sum_{j=1}^{i-2}\frac{1}{\lambda_{i-1}-\lambda_j}+\frac{1}{x_i-\lambda_{i-1}}+\sum_{j=i+1}^{n}\frac{1}{\lambda_j-\lambda_i}\right)+O(\varepsilon^2)\right)\\
&\leq \varepsilon^2\left(\sum_{j=1}^{i-2}\frac{1}{\lambda_{i-1}-\lambda_j}+\frac{1}{\lambda_i-\lambda_{i-1}-\varepsilon}+\sum_{j=i+1}^{n}\frac{1}{\lambda_j-\lambda_i}\right)+O(\varepsilon^3)\\
&\leq c\varepsilon^2.
\end{aligned}
$$

Similarly, we can show that $y_i^{(k)}$ converges to λ_i monotonically and quadratically. ■

To compute $p(\mu_i)$, where μ_i equals either $x_i^{(k)}$ or $y_i^{(k)}$, the same technique we used to isolate eigenvalues may be used. That is, we first reduce $(A - \mu_i M)$ to a tridiagonal matrix, then compute the Sturm sequences. The complexity of the reduction is of $O(n^2k)$, where k is the bandwidth. However, in this way, both the value of $p(\mu_i)$ and the number of eigenvalues which are less than μ_i are computed. Since all eigenvalues are isolated, only the value of $p(\mu_i)$ is needed. To evaluate $p(\mu_i)$, we may first compute the QR decomposition of $A - \mu_i M$. This can be done by the Givens transformations. Namely, compute orthogonal matrices $Q_1, Q_2, ..., Q_s$, where $s = nk - (k^2 - k)/2$, the number of elements under the diagonal, by the Givens transformations such that:

$$Q_s \cdots Q_1(A - \mu_i M) = R,$$

where R is an upper triangular matrix. Assume the diagonal elements of R are $\xi_1, \xi_2, ..., \xi_n$, then:

$$p(\mu_i) = det(A - \mu_i M)/\alpha = \frac{1}{\alpha}\prod_{i=1}^{n}\xi_i. \tag{8}$$

The complexity of the decomposition is only $O(nk^2)$.

From (8), (6) can be rewritten as:

$$
\begin{aligned}
x_i^{(k+1)} &= x_i^{(k)} - \frac{\prod_{j=1}^{n}\xi_j}{\alpha(x_i^{(k)}-x_1^{(k)})\cdots(x_i^{(k)}-x_{i-1}^{(k)})(x_i^{(k)}-y_{i+1}^{(k)})\cdots(x_i^{(k)}-y_n^{(k)})}\\
&= x_i^{(k)} - \frac{\xi_i\prod_{\substack{j=1\\ j\neq i}}^{n}\xi_j}{\alpha(x_i^{(k)}-x_1^{(k)})\cdots(x_i^{(k)}-x_{i-1}^{(k)})(x_i^{(k)}-y_{i+1}^{(k)})\cdots(x_i^{(k)}-y_n^{(k)})}\\
&= x_i^{(k)} - \frac{\xi_i}{\alpha}\prod_{\substack{j=1\\ j\neq i}}^{n}\frac{\xi_j}{\eta_{ij}}\\
i &= 1, 2, \cdots, n,
\end{aligned} \tag{9}
$$

where $\eta_{ij} = x_i^{(k)} - x_j^{(k)}$, if $j < i$ and $\eta_{ij} = x_i^{(k)} - y_j^{(k)}$, if $j > i$.

The computation of those products in (10) may suffer severe overflow and underflow problems. To avoid these problems, we group ξ_j/η_{ij}, $j \neq i$, into two groups. The absolute value of each element is less than one in the first group, and bigger than one in the second group. Then we multiply one element from the first group to one element in the second group. In this way, overflow and underflow problems can be avoided.

3. NUMERICAL RESULTS

In this section, we present our numerical results.

Definition 3.1 *The number*

$$c(A,B) = \min_{\|x\|_2=1} \left[(x^{\mathsf{T}}Ax)^2 + (x^{\mathsf{T}}Bx)^2\right]$$

is called the Crawford number *of the pencil* (A, B).

Theorem 3.1 *[6] Suppose* (A, B) *is an* $n \times n$ *symmetric definite pencil with eigenvalues*

$$\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n.$$

Suppose δA *and* δB *are symmetric* $n \times n$ *matrices satisfying*

$$\epsilon^2 = \|\delta A\|_2^2 + \|\delta B\|_2^2 < c(A,B).$$

Then the pencil $(A + \delta A, B + \delta B)$ *is a symmetric definite pencil with eigenvalues* $\mu_1 \leq \mu_2 \leq \cdots \leq \mu_n$ *which satisfy*

$$|\arctan(\lambda_i) - \arctan(\mu_i)| \leq \arctan\left(\frac{\epsilon}{c(A,B)}\right)$$

for $i = 1, 2, \cdots, n$.

Experiment 1. We implemented our algorithm LLGB and the routine RSG in EISPACK [14] on Equation (1). The matrices A and M are obtained from piecewise linear finite element discretization [3] of the Sturm–Liouville problem

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

where $u = u(x)$, $0 < x < \pi$, $u(0) = u'(\pi) = 0$, and $p(x) > 0$. When $[0, \pi]$ is divided into $n+1$ subintervals of equal length, Equation (1) is obtained. Here, both A and M are symmetric tridiagonal, and positive definite. To make the problem simpler, we use $p(x) = 1$ and $q(x) = 6$.

The experiment was conducted on a Sun-1000 workstation with IEEE standard and machine precision $\varepsilon \approx 2.2 \times 10^{-16}$.

For the symmetric narrow banded generalized eigenproblem, the subroutine RSG in EISPACK is somewhat less attractive since it transforms the generalized eigenproblem $Ax = \lambda Mx$ to a standard eigenproblem $\tilde{T}y = \lambda y$ of a dense matrix $\tilde{T}$. It can not take advantage of the banded structure of the pencil (A, M) and clearly requires $O(n^3)$ flops. We conduct the comparison here mainly because it is the only algorithm available in EISPACK for eigenproblems of symmetric definite pencils.

Table 1 shows the results of our algorithm LLGB, the algorithm RSG, and the BIS, the bisection method. The bisection method is a fully parallel method and also can take full advantage of the banded form of the matrices. The speed of the BIS is much faster that the RSG, though it is slower than our method. The reason is that, after the eigenvalues are isolated, our method converges quadratically, while the BIS still converges linearly. The execution time apparently shows that the complexities of LLGB and RSG are $O(n^2)$ and $O(n^3)$ respectively for the generalized tridiagonal problem.

Experiment 2. Let A be the Toeplitz matrix $[1, 4, 1]$ and M be the Toeplitz matrix $[10^{-14}, 2 \times 10^{-14}, 10^{-14}]$ with m_{11} and m_{nn} changed to 1. Although the condition number of M is about 10^{14}, the Crawford number

Table 1: **Execution Time (seconds) of Computed Eigenvalues.**

Order	Execution Time of All Eigenvalues		
n	RSG	LLGB	BIS
100	2.25	0.31	0.75
200	19.02	1.21	2.89
300	70.29	2.79	6.33
400	194.71	4.87	11.46

$c(A, M)$ is at least 2. So the eigenproblem of the pencil (A, M) is well-conditioned according to Theorem 3.1. The approximate eigenvalues of (A, M) evaluated by LLGB and RSG are denoted by $\{z_i\}_{i=1}^n$ and $\{r_i\}_{i=1}^n$ respectively.

To compare the accuracy of LLGB and RSG on (A, M), we use the eigenvalue counter $\kappa(\lambda)$ to conduct bisection and obtain intervals $\{[a_i, b_i]\}_{i=1}^n$ such that $\kappa(a_i) \leq i - 1$, $\kappa(b_i) \geq i$ and $|\arctan(b_i) - \arctan(a_i)| \leq \varepsilon$. Notice that

$$\sigma = \sqrt{(2.51\varepsilon\|A\|_\infty)^2 + (3.51\varepsilon\|M\|_\infty)^2} < 16\varepsilon.$$

So the error bound in Theorem 3.1 is $\frac{\sigma}{c(A,M)} < 8\varepsilon$. Let $[\hat{a}_i, \hat{b}_i] \supset [a_i, b_i]$ be the interval for which

$$|\arctan(\hat{a}_i) - \arctan(a_i)| = |\arctan(\hat{b}_i) - \arctan(b_i)| = 8\varepsilon.$$

Then, the i^{th} smallest exact eigenvalue λ_i of the pencil (A, M) is in $[\hat{a}_i, \hat{b}_i]$. Let $\tilde{\lambda}_i = (a_i + b_i)/2$. Then

$$\begin{aligned} |\arctan(\tilde{\lambda}_i) - \arctan(\lambda_i)| &\leq |\arctan(a_i) - \arctan(b_i)| + |\arctan(\hat{a}_i) - \arctan(a_i)| \\ &\leq 9\varepsilon < 2 \times 10^{-15}. \end{aligned}$$

So if x is considered an approximate value of λ_i and $|x - \tilde{\lambda}_i| = \delta$, then

$$\delta - 2 \times 10^{-15} \leq |x - \lambda_i| \leq \delta + 2 \times 10^{-15}.$$

Let z_i's and r_i's be numerical eigenvalues obtained by LLGB and RSG respectively. Table 2 lists errors

$$\max_{1\leq i\leq n} \{|\arctan(z_i) - \arctan(\lambda_i)|\} \quad \text{and} \quad \max_{1\leq i\leq n} \{|\arctan(r_i) - \arctan(\lambda_i)|\}$$

for different matrix size n. The data clearly indicate that our algorithm is very accurate regardless of the condition number of matrix M, while RSG fails to compute some of the eigenvalues when M is ill-conditioned.

Experiment 3. Our algorithm LLGB and the routine RGG are applied to Equation (1) where matrices A and M are obtained from the finite difference discretizations of the following problem:

$$L_4u - \lambda L_2u = 0,$$

Table 2: **Accuracy Comparison Between LLGB and RSG.**

Order	Error of LLGB	Error of RSG
n	$\max_{1\le i\le n} \lvert\arctan(z_i) - \arctan(\lambda_i)\rvert$	$\max_{1\le i\le n} \lvert\arctan(r_i) - \arctan(\lambda_i)\rvert$
5	$\le 6.3\times10^{-15}$	$\le 2.2\times10^{-15}$
10	$\le 7.2\times10^{-15}$	$\ge 2.3\times10^{-7}$
20	$\le 5.8\times10^{-15}$	$\ge 2.5\times10^{-4}$
50	$\le 4.3\times10^{-15}$	≥ 1.07

Table 3: **Average Execution Time (seconds) of Computed Eigenvalues.**

Order	Execution Time of All Eigenvalues			
	five-diagonal		seven-diagonal	
n	RGG	LLGB	RGG	LLGB
60	3.4	2.7	3.6	3.1
121	24.3	17.2	27.1	21.6
180	77.4	45.6	85.6	71.8
241	183.9	103.3	214.2	165.7

where

$$L_4u = \sum_{k=0}^{2}(-1)^k \frac{d^k}{dx^k}(p_k(x)\frac{d^ku}{dx^k}),$$

and

$$L_2u = \sum_{k=0}^{1}(-1)^k \frac{d^k}{dx^k}(q_k(x)\frac{d^ku}{dx^k}),$$

with

$$u(0) = u'(0) = u(1) = u'(1) = 0,$$

and $p_k(x) = 1$ and $q_k(x) = x$.

We use five-point finite difference schemes and Richardson's extrapolation to get the matrices A and M with 5, and 7 nonzero diagonals, respectively.

The subroutine RGG in EISPACK is less sensitive to the conditioning of M. However, it can not take advantage of the symmetric and banded structure of A and M.

Table 3 shows the results of our algorithm LLGB and the algorithm RGG. It appears that our algorithm leads in speed by a considerable margin in comparison with the RGG when bandwidth is small.

REFERENCES

[1] A.E. Berger, J.M. Solomon, and M. Ciment, "Higher Order Accurate Tridiagonal Difference Methods for Diffusion Convection Equations", in *Advances in Computer Methods for Partial Differential Equations: III.* ed. R. Vichnevetsky and R.S. Stepleman. IMACS, 1978, pp. 322–330.

[2] B. Grieves and D. Dunn, "Symmetric Matrix Methods for Schrödinger Eigenvectors", *J. Phys. A: Math Gen.*, **23** (1980), pp. 5479–5491.

[3] G. Strang and G. Fix, *An Analysis of the Finite Element Method.* Englewood Cliffs, NJ: Prentice-Hall, 1973.

[4] W. Weaver, Jr. and P.R. Johnson, *Finite Elements for Structural Analysis.* Englewood Cliffs, NJ: Prentice-Hall, 1984.

[5] H.B. Keller, *Numerical Methods for Two-Point Boundary Value Problems.* Blaisdell Publishing Inc., 1968.

[6] G.H. Golub and Van Loan, *Matrix Computations.* Baltimore, MD: The Johns Hopkins University Press, 1989.

[7] B.N. Parlett, *The Symmetric Eigenvalue Problem.* Englewood, Cliffs, NJ: Prentice-Hall, 1980.

[8] S-S. Lo, B. Phillipe, and A. Sameh, "A Multiprocessor Algorithm for the Symmetric Tridiagonal Eigenvalue Problems", *SIAM J. Sci. Stat. Comput.*, **8** (1987), pp. 155–165.

[9] K. Li, "A Fully Parallel Method for Tridiagonal Eigenvalue Problem", *Internat. J. of Math. & Math. Sci.*, **17(4)** (1994), pp. 741–752.

[10] K. Li and T.Y. Li, "An Algorithm for Symmetric Tridiagonal Eigen-Problems — Divide and Conquer with Homotopy Continuation", *SIAM J. Sci. Comput.*, **14(3)** (1993), pp. 735-751.

[11] K. Li, T.Y. Li, and Z. Zeng, "An Algorithm for the Generalized Symmetric Tridiagonal Eigenvalue Problem", *Numerical Algorithms*, **8** (1994), pp. 269–291.

[12] J.H. Wilkinson, *The Algebraic Eigenvalue Problem.* Oxford: Oxford University Press, 1965.

[13] K. Li and T.Y. Li, "A Parallel Method for the Banded Eigenvalue Problem", *AJSE*, **22(2C)** (1997), pp. 31–40.

[14] B.S. Garbow *et al., Matrix Eigensystem Routines — EISPACK Guide Extension.* New York: Springer-Verlag, 1977.

Paper Received 30 September 1998; Accepted 27 February 2000.