# DURAND–KERNER ROOT–FINDING METHOD FOR THE GENERALIZED TRIDIAGONAL EIGENPROBLEM

Kuiyuan Li

**Abstract.** In this paper we present a method, parallel in nature, for finding all eigenvalues of a symmetric generalized tridiagonal matrix. Our method employs the determinant evaluation and the Durand–Kerner root–finding scheme. It will be shown that the method converges quadratically and is reliable, efficient, and easy to implement in practice.

**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 tridiagonal and one of them, say $M$, is positive definite. These assumptions imply real eigenvalues and the existence of an $M$ — orthogonal basis of eigenvectors. Eigenvalue problems with such a special structure arise in many applications, such as finding numerical solutions of the Sturm–Liouville and the radial Schrödinger equations [1, 3, 8, 18] and the finite element approximations for free longitudinal vibrations problems of non–uniform rod [16, 19].

There are some traditional approaches to solving (1). The first approach is to reduce (1) to a standard eigenvalue problem [7, 14, 20]:

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

where $M = LL^T$ is the Cholesky factorization of $M$. Then the eigenvalue problem (2) can be solved by many efficient algorithms, such as the QR algorithm [14], the bisection/multisection algorithms [13], the divide–conquer algorithm [4, 5], and homotopy algorithm [11]. However, this approach is less attractive because it can not take advantage of the tridiagonal 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 complexity of this method is $O(n^3)$.

The second approach, the QZ method [20], disregards the symmetry and the tridiagonal structure of the problem and after a direct phase requiring $O(n^3)$ operations, enters an iterative phase requiring $O(n^3)$ operations to simultaneously reduce $A$ and $M$ to triangular–quasi–triangular form. This algorithm is not sensitive to the conditioning of $M$, though it is even more expensive than the first approach.

Another approach is the bisection/multisection method [15]. This parallel method is very easy to implement in practice and is reliable and accurate. However, the speed of the convergence is slow.

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

$$p(\lambda) = \frac{1}{\alpha} \det(a - \lambda M), \tag{3}$$

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

Our method can be divided into two stages: root isolation and root extraction. The root isolation can be done by bisections or multisections of order $n_p$, the number of processors. A multisection splits a given interval $[a, b]$ into $n_p$ subintervals $[\mu_i, \mu_{i+1}]$, where $\mu_i = a + i(b - a)/n_p$, $i = 0, 1, 2, \ldots, n_p$, by computing Sturm sequences at $\mu_i$. As a result, the number of roots of $p(\lambda)$ which are smaller than each $\mu_i$ are obtained and so are the number of roots in each subinterval. The process is recursively applied to every subinterval containing more than one root. This step results in a list of intervals consisting of single roots. When multiple roots are present, the isolation process continues until the required precision is reached. The root extraction shall be done by the Durand–Kerner root–finding method which shall be discussed in detail in section 3. We shall also show how we can virtually eliminate the overflow and underflow problems in section 2 and 3. In section 4, we shall show that our method is reliable and backward stable.

There are other approaches to this problem. In [2, 17], divide–and–conquer methods are proposed; the method based on the determinant evaluation and the Laguerre iterations is suggested in [10, 12]; and the parallel implementations of the multisection method and Lanczos method are discussed in [15].

**2. The Sturm Sequences.** Without loss of generality, we assume the matrix pencil $(A, M)$ is unreducible, that is, $\beta_i^2 + \delta_i^2 \neq 0$, $i = 2, \ldots, n$, where

$$A = \begin{pmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{n-1} & \alpha_{n-1} & \beta_n \\ & & & \beta_n & \alpha_n \end{pmatrix} \text{ and } M = \begin{pmatrix} \gamma_1 & \delta_2 & & & \\ \delta_2 & \gamma_2 & \delta_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \delta_{n-1} & \gamma_{n-1} & \delta_n \\ & & & \delta_n & \gamma_n \end{pmatrix}. \tag{4}$$

It is well known that polynomials defined by

$$\begin{cases} \rho_0(\lambda) = 1, \quad \rho_1(\lambda) = \alpha_1 - \lambda\gamma_1, \\ \rho_i(\lambda) = (\alpha_i - \lambda\gamma_i)\rho_{i-1}(\lambda) - (\beta_i - \lambda\delta_i)^2\rho_{i-2}(\lambda), \quad i = 2, 3, \ldots, n, \end{cases} \tag{5}$$

form a Sturm sequence and

$$p(\lambda) = \frac{1}{\alpha}\rho_n(\lambda).$$

The number of sign changes between consecutive terms of $\{\rho_i\}_{i=1}^n$, denoted by $\kappa(\lambda)$ is equal to the number of roots of $p(\lambda)$ that are less than $\lambda$.

The Sturm sequence (5) may suffer from severe overflow and underflow problems and require frequent scaling [10]. To remedy this, we propose following alternative recurrences. Let

$$\xi_i = \frac{\rho_i}{\rho_{i-1}}, \quad i = 1, 2, \ldots, n. \tag{6}$$

Dividing both sides of (5) by $\rho_{i-1}$ yields,

$$\begin{cases} \xi_1 = \alpha_1 - \lambda\gamma_1, \\ \xi_i = \alpha_i - \lambda\gamma_i - \frac{(\beta_i - \lambda\delta_i)^2}{\xi_{i-1}}, \quad i = 2, 3, \ldots, n. \end{cases} \tag{7}$$

To avoid possible breakdown, namely, when $\xi_i = 0$ for certain $i$, the following adjustment is asserted:

- If $\xi_1 = 0$, set $\xi_1 = \gamma_1 \epsilon^2$.
- If $\xi_i = 0$, $i > 1$, set $\xi_i = \frac{(|\beta_i| + |\lambda \delta_i|)^2 \epsilon^2}{\xi_{i-1}}$.

That is, if $\xi_i = 0$, we perturb the corresponding entries beyond the last significant digit stored in the machine.

Obviously,

$$\rho_i = \prod_{k=1}^{i} \xi_k \tag{8}$$

and $\kappa(\lambda)$, the number of roots of $p(\lambda)$ that are less than $\lambda$, is now equal to the number of negative terms in $\{\xi_i\}_{i=1}^{n}$.

(7) virtually eliminates hazards of overflow and underflow problems to compute $\kappa(\lambda)$ because of its self–scaling.

**3. The Durand–Kerner Method.** Since

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

it can be rewritten as

$$p(\lambda) = \lambda^n + a_1 \lambda^{n-1} + \cdots + a_{n-1} \lambda + a_n. \tag{9}$$

Assume $\lambda_1, \lambda_2, \ldots, \lambda_n$ are zeroes of $p(\lambda)$.

Let

$$h(\lambda) = (\lambda - \mu_1)(\lambda - \mu_2) \cdots (\lambda - \mu_n).$$

The following iterative formula

$$\mu_i^{(k+1)} = \mu_i^{(k)} - \frac{p(\mu_i^{(k)})}{h'(\mu_i^{(k)})}, \quad i = 1, 2, \ldots, n \tag{10}$$

is called the Durand–Kerner root–finding iteration [6, 9]. Numbers $\mu_i^{(k)}$, $i = 1, 2, \ldots, n$, are a set of approximations to $n$ zeroes of $p(\lambda)$. Numbers $\mu_i^{(k+)}$, $i = 1, 2, \ldots, n$, are a set of, normally, better approximations.

Theorem 3.1. Let $\mu_i^{(0)}$, $i = 1, 2, \ldots, n$ be arbitrary real numbers and $\mu_i^{(1)}$, $i = 1, 2, \ldots, n$ be computed from (10). Then

$$\sum_{i=1}^{n} \mu_i^{(1)} = \sum_{i=1}^{n} \lambda_i, \tag{11}$$

where $\lambda_i$'s are eigenvalues.

Proof. Since $\lambda_1, \lambda_2, \ldots, \lambda_n$ are roots of $p(\lambda) = 0$,

$$p(\lambda) = (\lambda - \lambda_1)(\lambda - \lambda_2)\cdots(\lambda - \lambda_n). \tag{12}$$

Comparing (9) with (12) yields

$$a_1 = -\sum_{i=1}^{n} \lambda_i.$$

On the other hand, by the results from [6, 9],

$$a_1 = -\sum_{i=1}^{n} \mu_i^{(1)}.$$

Therefore,

$$\sum_{i=1}^{n} \mu_i^{(1)} = \sum_{i=1}^{n} \lambda_i.$$

Theorem 3.1 implies that after one iteration, the sum of $n$ approximations is equal to the sum of all roots of $p(\lambda)$, even if initial approximations are arbitrary. The sum of $n$ improved approximations is always constant and equal to the sum of all roots of $p(\lambda)$.

It can be shown [6] that the iteration (10) is the Newton method applied to the function $f(x) = (f_1(x), f_2(x), \ldots, f_n(x))^T$, where $x = (x_1, x_2, \ldots, x_n)^T$ and

$$
\begin{aligned}
f_1(x) &= x_1 + x_2 + \cdots + x_n + a_1, \\
&\vdots \\
f_k(x) &= \sum_{i_1 < i_2 < \cdots < i_k} x_{i_1} x_{i_2} \cdots x_{i_k} + (-1)^{k-1} a_k, \\
&\vdots \\
f_n(x) &= \prod_{i=1}^{n} x_i + (-1)^{n-1} a_n.
\end{aligned}
$$

The Newton iteration implies that the convergence is quadratic whenever the Jacobian of $f(x)$ is nonsingular at $(\lambda_1, \lambda_2, \ldots, \lambda_n)^T$, i.e., whenever those roots $\lambda_i$ are simple and there are $r_i$ components of $x$ which converge to $\lambda_i$, if the multiplicity of $\lambda_i$ is $r_i$. In the case of multiple zeroes, the convergence is only linear. However, the case of multiple zeroes can be eliminated since our method consists of two parts: root isolation and root extraction. Without loss of generality, assume $\lambda_1$ is a multiple root with multiplicity $m$ and $\lambda_{m+1}, \ldots, \lambda_n$ are distinct. After the root isolation, $\lambda_1$ is obtained. Therefore, we use the modified Durand–Kerner method:

$$
\mu_i^{(k+1)} = \lambda_1, \quad i = 1, 2, \ldots, m.
$$

$$
\mu_i^{(k+1)} = \mu_i^{(k)} - \frac{p(\mu_i^{(k)})}{h'(\mu_i^{(k)})}, \quad i = m+1, m+2, \ldots, n.
$$

The convergence is still quadratic.

Although (10) always converges in practice [6, 9] and no counterexample has been given so far, no proof of this property has been given. Many authors conjecture that this scheme converges for almost any starting points. However, to our special problem, we have the following result.

Theorem 3.2. Assume

$$\mu_1 < \lambda_1 < \mu_2 < \lambda_2 < \cdots < \mu_{[\frac{n}{2}]} < \lambda_{[\frac{n}{2}]} < \lambda_{[\frac{n}{2}]+1} < \mu_{[\frac{n}{2}]+1} < \cdots < \lambda_n < \mu_n,$$

where $\lambda_1, \lambda_2, \ldots, \lambda_n$ are roots of $p(\lambda) = 0$ and $\mu_1, \mu_2, \ldots, \mu_n$ are starting points for (10). Then,

$$\mu_i < \mu_i^{(1)}, \quad \text{if } i \leq \frac{n}{2}$$

and

$$\mu_i > \mu_i^{(1)}, \quad \text{if } i > \frac{n}{2},$$

where $\mu_1^{(1)}, \mu_2^{(1)}, \ldots, \mu_n^{(1)}$ are computed from (10).

Proof.

$$\begin{aligned}
\mu_i^{(1)} &= \mu_i - \frac{p(\mu_i)}{h'(\mu_i)} \\
&= \mu_i - \frac{\prod_{j=1}^{n} (\mu_i - \lambda_j)}{\prod_{\substack{j=1 \\ j \neq i}} (\mu_i - \mu_j)} \\
&= \mu_i - (\mu_i - \lambda_i) \frac{\prod_{\substack{j=1 \\ j \neq i}} (\mu_i - \lambda_j)}{\prod_{\substack{j=1 \\ j \neq i}} (\mu_i - \mu_j)}.
\end{aligned}$$

Clearly,

$$\frac{\prod_{\substack{j=1 \\ j \neq i}} (\mu_i - \lambda_j)}{\prod_{\substack{j=1 \\ j \neq i}} (\mu_i - \mu_j)} > 0, \quad \text{for } i = 1, 2, \ldots, n,$$

therefore,

$$\mu_i < \mu_i^{(1)}, \quad \text{if } i \leq \frac{n}{2}$$

and

$$\mu_i > \mu_i^{(1)}, \quad \text{if } i > \frac{n}{2}.$$

Theorem 3.1 and 3.2 suggest that we should choose starting points like those we mentioned in Theorem 3.2. After each iteration, we should have better approximations. In case $\mu_i^{(k+1)} > \mu_{i+1}^{(k)}$ happens for $i \leq \frac{n}{2}$ or $\mu_i^{(k+1)} < \mu_{i-1}^{(k)}$ for $i \geq \frac{n}{2}$, we give up $\mu_i^{(k+1)}$ and do bisection at this point to insure the convergence.

From (8), (10) can be rewritten as

$$\begin{aligned}
\mu_i^{(k+1)} &= \mu_i^{(k)} - \frac{\prod_{j=1}^n \xi_j}{\alpha \prod_{\substack{j=1 \\ j\neq i}}^n (\mu_j^{(k)} - \mu_i^{(k)})} \qquad (13) \\
&= \mu_i^{(k)} - \frac{\xi_i \prod_{\substack{j=1 \\ j\neq i}}^n \xi_j}{\alpha \prod_{\substack{j=1 \\ j\neq i}}^n \eta_{ji}} \\
&= \mu_i^{(k)} - \frac{\xi_i}{\alpha} \prod_{\substack{j=1 \\ j\neq i}}^n \frac{\xi_j}{\eta_{ji}} \\
& i = 1, 2, \ldots, n,
\end{aligned}$$

where $\eta_{ji} = \mu_j^{(k)} - \mu_i^{(k)}$.

Directly computing those products in (13) may lead to severe overflow and underflow problems. To avoid the problems, we do the following:

(1). Compute $v_j = \frac{\xi_j}{\eta_{ji}}$, $j = 1, 2, \ldots, n$ and $j \neq i$.

(2). If $|v_j| \leq 1$, let $v_j$ in $S$. If $|v_j| > 1$, let $v_j$ in $B$. Then $S = \{s_1, s_2, \ldots\}$ and $B = \{b_1, b_2, \ldots\}$.

(3). Let $w = s_1 b_1 s_2 b_2 \cdots$.

Hence,

$$\mu_i^{(k+1)} = \mu_i^{(k)} - w\xi_i/\alpha, \quad i = 1, 2, \ldots, n. \qquad (14)$$

In this way, overflow and underflow problems can be eliminated.

Since $\mu_i^{(k+1)}$ and $\mu_j^{(k+1)}$, $i \neq j$, can be computed independently, the method is parallel in nature.

**4. Error Analysis.** Let $fl(\bullet)$ denote the floating point computation of $\bullet$. In practice, when we intend to evaluate zeros of $p(\lambda)$, actually zeros of $fl[p(\lambda)]$ are evaluated. We now do some error analysis and thereby establish the backward stability of our algorithm.

The most frequently used model for floating point arithmetic is

$$fl(x \circ y) = (x \circ y) \cdot (1 + e), \quad |e| \leq \epsilon,$$

where $\circ$ is either $+, -, \times$, or $/$ and $\epsilon$ above is the machine precision, which is approximately $6 \times 10^{-8}$ for single precision and $2.2 \times 10^{-16}$ for double precision under IEEE standards.

Proposition 4.1. [12]

$$fl[f(\lambda)] = (1 + \epsilon_1) \det[(A + \delta A) - \lambda(M + \delta M)], \tag{15}$$

where

$$|\epsilon_1| \leq n\epsilon,$$

and both $\delta A$ and $\delta M$ are symmetric tridiagonal matrices satisfying entrywise inequalities

$$|\delta A| \leq 2.51\epsilon|A|, \quad |\delta M| \leq 3.51\epsilon|M|. \tag{16}$$

Assume $\lambda_{min}(M) > 3.51\epsilon\|M\|_\infty$. Then by Proposition 4.1,

$$\begin{aligned} \lambda_{min}(M + \delta M) &> \lambda_{min} - \|\delta M\|_\infty \\ &\geq \lambda_{min} - 3.51\epsilon\|M\|_\infty > 0. \end{aligned}$$

Therefore, $(A + \delta A, M + \delta M)$ is a symmetric, positive definite, tridiagonal pencil, if $\lambda_{min}(M) > 3.51\epsilon\|M\|_\infty$ and this implies that, unless the pencil $(A, M)$ is extremely ill–conditioned, for which $\lambda_{min} < 3.51\epsilon\|M\|_\infty$, the floating point computation of our algorithm is still performed on a symmetric definite pencil.

This establishes the backward stability of our algorithm.

**5. Conclusion.** In this paper, we have shown that the Durand–Kerner root–finding method converges for the generalized tridiagonal eigenvalue problem. The asymptotic convergence rate is quadratic. After each iteration, the sum of the improved approximations is the sum of all the eigenvalues. This indicates why our method converges so fast in practice. We have also shown how overflow and underflow problems can be eliminated. These good properties and schemes make our method an excellent candidate for the generalized tridiagonal eigenvalue problem.

## *References*


1. A. E. Berger, J. M. Solomon, and M. Ciment, "Higher Order Accurate Tridiagonal Difference Methods for Diffusion Convection Equations," *Advances in Computer Methods for Partial Differential Equations III*, R. Vichnevetsky and R. S. Stepleman, editors, IMACS, 1978, 322–330.
2. C. F. Borges and W. B. Gragg, *A Parallel Divide and Conquer Algorithm for the Generalized Real Symmetric Definite Tridiagonal Eigenproblem*, Naval Postgraduate School, preprint, 1992.
3. R. F. Boisvert, "Families of Higher Order Accurate Discretizations of Some Elliptic Problems," *SIAM J. Sci. Stat. Comput.*, 2 (1981), 268–284.
4. J. J. M. Cuppen, "A Divide and Conquer Method for the Symmetric Tridiagonal Eigenproblem," *Numer. Math.*, 36 (1981), 177–195.
5. J. J. Dongarra and D. C. Sorensen, "A Fully Parallel Algorithm for the Symmetric Eigenvalue Problem," *SIAM J. Sci. Stat. Comput.*, 8 (1987), 139–154.
6. P. Fraigniaud, "The Durand–Kerner Polynomials Roots–Finding Method in Case of Multiple Roots," *BIT*, 31 (1991), 112–123.
7. G. H. Golub and Van Loan, *Matrix Computations*, 2nd ed., The Johns Hopkins University Press, Baltimore, Maryland, 1989.
8. B. Grieves and D. Dunn, "Symmetric Matrix Methods for Schrödinger Eigenvectors," *J. Phys. A: Math Gen.*, Vol. 23, 5479–5491.
9. G. Kjellberg, "Two Observations on Durand–Kerner's Root–Finding Method," *BIT*, 24 (1984), 556–559.
10. K. Li, "A Fully Parallel Method for Tridiagonal Eigenvalue Problem," *Internat. J. of Math. & Math. Sci.*, 17 (1994), 741–752.

11. K. Li and T. Y. Li, "An Algorithm for Symmetric Tridiagonal Eigen–Problems – Divide and Conquer with Homotopy Continuations," *SIAM J. Sci. Comput.*, 14 (1993), 735–751.

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

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

14. B. N. Parlett, *The Symmetric Eigenvalue Problem*, Prentice–Hall, Inc., Englewood Cliffs, New Jersey, 1980.

15. B. Philippe and B. Vital, "Parallel Implementations for Solving Generalized Eigenvalue Problems with Symmetric Sparse Matrices," *Applied Numerical Mathematics*, 12 (1993), 391–402.

16. Y. M. Ram and G. M. L. Gladwell, "Constructing a Finite–Element Model of a Vibratory Rod From Eigendata," *J. of Sound and Vibration*, 165 (1993), 36–65.

17. C. J. Ribbens and C. Beattie, "Parallel Solutions of Generalized Symmetric Tridiagonal Eigenvalue Problem on Shared Memory Multiprocessors," *Technical Report TR 92-47*, Department of Computer Science, Virginia Polytechnic Institute and State University, 1992.

18. G. Strang and G. Fix, *An Analysis of the Finite Element Method*, Prentice–Hall, Inc., Englewood Cliffs, New Jersey, 1973.

19. W. Weaver, Jr. and P. R. Johnson, *Finite Elements for Structural Analysis*, Prentice–Hall, New Jersey, 1984.

20. J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Oxford University Press, Oxford, 1965.

Kuiyuan Li
Department of Mathematics and Statistics
University of West Florida
Pensacola, FL 32514
email: kli@math-stat.math.uwf.edu