Parallel Eigenvalue Algorithms for Symmetric Circulant Tridiagonal and Symmetric Quindiagonal Matrices *

Wen-Yan Tian and Chang-Biau Yang
Department of Applied Mathematics
National Sun Yat-sen University
Kaohsiung, Taiwan 80424
cbyang@math.nsysu.edu.tw

Abstract

In this paper, we propose two parallel algorithms to calculate the eigenvalues of two special matrices, the symmetric circulant tridiagonal matrix and the symmetric quindiagonal matrix. Our algorithms are based on the bisection method and the Sylvester's law of inertia. For the symmetric circulant tridiagonal matrix, we are the first one to solve the eigenvalue problem in parallel. The basic work in the algorithm is to compute some linear recurrence sequences and the time required in one iteration is $O(\log n)$ by using $O(n/\log n)$ processors. For the symmetric quindiagonal matrix, we use the concept of block submatrices to solve it. We get similar recurrence forms. Though they are not linear, they are simpler than the previous algorithm based upon the matrix determinant.

Key words. eigenvalue, symmetric circulant tridiagonal matrix, symmetric quindiagonal matrix.

1 Introduction

The problem of finding the eigenvalues of a matrix becomes increasingly important in the sciences and engineering areas, such as the control theory, mechanical calculations, structural analysis, heat conduction, and wave propagation [12]. These problems can be represented quantitatively by *differential equations or partial differential equations*, and solving these differential equations can be transformed into matrix eigenvalue problems. The eigenvalue problem can be formulated as

$$Ax = \lambda x,$$

where $A \in R^{n\times n}$ and λ is the eigenvalue of A with the associated eigenvector x. In general, there are n eigenvalues for A. That is, $\lambda = (\lambda_1, \lambda_2, \cdots, \lambda_n)$ is desired to be calculated.

The generalized eigenvalue problem [9]

$$Ax = \lambda Bx, \quad where \quad A, B \in C^{n\times n}$$

can be used to solve the wave function of quantum mechanical methods in chemical and physical properties of molecules and macromolecules. Eigenvalues of the problem can be used to determine the relation of natural frequencies between building structure and earthquake [14]. Moreover, if B is nonsingular, then $\lambda(A, B) = \lambda(B^{-1}A, I)$, where $\lambda(A, B)$ denotes the set of eigenvalue of matrices A and B. That is, the generalized eigenvalue problem can also be transformed into an eigenvalue problem of the form

$$Cx = \lambda x, \quad where \quad C = B^{-1}A.$$

Then we compute the eigenvalues of C.

In the science and engineering areas, most applications require the eigenvalues of a large matrix. Especially, symmetric tridiagonal systems [4, 8, 10, 13, 15], banded systems [9], and circulant tridiagonal systems [3, 11] are frequently discussed. For instance, in computer aided design, graphic pattern recognition and image processing, closed curve fitting [2] is important and it can be transformed into a circulant tridiagonal system.

In the past, there are many algorithms in solving the matrix eigenvalue problems, including the power method, the QR method, the Jacobi method, the QZ method, the Cuppen's divide and conquer method [4], the bisection method [1], and so on.

Sometimes, we are not interested in all eigenvalues of a matrix, and only need some specified eigenvalues, such as the smallest or largest one. The bisection method is suitable for finding one or some specified eigenvalues of a matrix. In this paper, we shall propose parallel algorithms based on the bisection method and the Sylvester's law of inertia [6] to calculate the

*This research work was partially supported by the National Science Council of the Republic of China under contract NSC 86-2215-M-110-010.

eigenvalues of the real circulant symmetric tridiagonal matrix and the real symmetric quindiagonal matrix. Here, all the matrices we discussed are real matrices, and the entries of a matrix we do not write out are all zero entries. In the former algorithm, the time required in one iteration is $O(\log n)$ by using $O(n/\log n)$ processors. In the latter algorithm, it applies the concept of block submatrices. The first half of the algorithm has to calculate $O(n)$ submatrices of size 2×2 sequentially. Then all eigenvalues of these submatrices can be calculated in parallel in constant time. Our algorithm is simpler than the previous algorithm based upon the matrix determinant [5], which is also an sequential algorithm.

The rest of this paper is organized as follows. In Section 2, we shall present Sylvester's law of inertia, a valuable property for solving the eigenvalue problem, and the bisection method, a famous one to solve the eigenvalue problem. In Section 3, we shall propose a parallel algorithm, based upon the bisection method, to calculate the eigenvalues of the symmetric circulant tridiagonal matrix. In Section 4, an algorithm will be proposed to calculate the eigenvalues of the symmetric quindiagonal matrix. In Section 5, we analyze the time complexities of our algorithms. And finally, the conclusion will be given in Section 6.

2 The Bisection Method

In this section, we will briefly review the *bisection method* [1], for solving the eigenvalue problem on the real symmetric tridiagonal matrix. The bisection method, based on the *Sylvester's law of inertia* and also known as the *slicing* or *Sturm sequence method*, is a simple and inexpensive procedure for calculating the eigenvalues, in a given interval, of a real symmetric tridiagonal matrix. Each iteration converges approximately half in the given interval. As we shall see, it is suitable for finding one or more eigenvalues in a specified interval.

The *Gerschgorin's theorem* [6] shows that each eigenvalue of matrix $A \in R^{n\times n}$ lies in a certain interval

$$|\lambda_i - a_{ii}| \le r_i, \quad 1 \le i \le n,$$

where λ_i is the eigenvalue of A and $r_i = \sum_{j=1,j\ne i}^{n} |a_{ij}|$. So, all eigenvalues of A lie between the interval $[min_{1\le i\le n}(a_{ii} - r_i),\ max_{1\le i\le n}(a_{ii} + r_i)]$. The eigenvalues of a symmetric matrix $A \in R^{n\times n}$ are all real. Let $v(A)$, $\zeta(A)$ and $\pi(A)$ denote the number of negative, zero, and positive eigenvalues of A, respectively. Then the order triple $(v(A), \zeta(A), \pi(A))$ is called the *inertia* of A. Let $\lambda_1, \lambda_2, \ldots, \lambda_n$ be the eigenvalues of A, ordered so that $\lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_n$. Suppose that for any given $\sigma \in R$, we can determine the inertia of $A - \sigma I$. The number $\pi(A - \sigma I)$ equals the number of eigenvalues of A that are greater than σ. In other words,

If $\pi(A - \sigma I) = i$, with $1 \le i \le n$ then
$\lambda_n \le \cdots \le \lambda_{i+1} \le \sigma \le \lambda_i \le \cdots \le \lambda_1$.

This splits the set of eigenvalues of A into two subsets. Repeatedly slicing the set of eigenvalues by choosing proper values of σ, we can determine all the eigenvalues of matrix A with great precision.

Next, we shall see how Sylvester's law is applied to slice the set of eigenvalues into subsets.

Theorem 1 *(Sylvester's law of inertia) [6]*
If $A \in R^{n\times n}$ is a symmetric matrix and $X \in R^{n\times n}$ is nonsingular, then A and XAX^T have the same inertia, that is, $v(A) = v(XAX^T)$, $\zeta(A) = \zeta(XAX^T)$, and $\pi(A) = \pi(XAX^T)$.

The *similarity transformation* [6] is to transform the original matrix into either a triangular matrix or a near-triangular matrix form of which the eigenvalues can be easily calculated. Two matrices $A, B \in R^{n\times n}$ is said to be *similar* if there exists a nonsingular matrix $U \in R^{n\times n}$ such that $B = U^{-1}AU$. And because two similar matrices have the same eigenvalues, so that matrices A and B have the same eigenvalues. A procedure often used to solve the eigenvalue problem is to reduce the original $full$ matrix A to a tridiagonal matrix B by using the similarity transformation. Thus, finding the eigenvalues of a symmetric tridiagonal matrix is crucial in the process of finding the eigenvalues of symmetric matrices.

Now, we shall describe how the bisection method to solve the eigenvalues of the real symmetric tridiagonal matrix

$$A = \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{n-2} & \alpha_{n-1} & \beta_{n-1} \\ & & & \beta_{n-1} & \alpha_n \end{bmatrix}.$$

Suppose we know there is an eigenvalue in some specified interval $(a, b]$. We want to calculate the eigenvalue with error less than some specified tolerance $\epsilon > 0$. We can compute $\pi(A - \sigma I)$ by decomposing $A - \sigma I$ as

$$A - \sigma I = L_\sigma D_\sigma L_\sigma^T$$

where L_σ is unit lower triangular and D_σ is diagonal. Here, a unit lower triangular matrix is a lower triangular matrix with all entries on the diagonal being 1. By Sylvester's law of inertia, $\pi(A - \sigma I)$ is equal to $\pi(D_\sigma)$. Since D_σ is diagonal, $\pi(D_\sigma)$ is equal to the number of positive elements on the main diagonal of D_σ. The decomposition $A - \sigma I = L_\sigma D_\sigma L_\sigma^T$ can be written more explicitly as

$$A - \sigma I = \begin{bmatrix} \alpha_1 - \sigma & \beta_1 & & & \\ \beta_1 & \alpha_2 - \sigma & \beta_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{n-2} & \alpha_{n-1} - \sigma & \beta_{n-1} \\ & & & \beta_{n-1} & \alpha_n - \sigma \end{bmatrix}$$

$$= \begin{bmatrix} 1 & & & & \\ l_1 & 1 & & & \\ & l_2 & 1 & & \\ & & \ddots & \ddots & \\ & & & l_{n-1} & 1 \end{bmatrix} \begin{bmatrix} d_1 & & & & \\ & d_2 & & & \\ & & \ddots & & \\ & & & d_{n-1} & \\ & & & & d_n \end{bmatrix}$$

$$\begin{bmatrix} 1 & l_1 & & & \\ & 1 & l_2 & & \\ & & \ddots & \ddots & \\ & & & 1 & l_{n-1} \\ & & & & 1 \end{bmatrix}.$$

It is a simple matter to carry out the matrix multiplications on the right-hand side and we can get

$$\begin{aligned} \alpha_1 - \sigma &= d_1, \\ \beta_i &= l_i d_i, \\ \alpha_{i+1} - \sigma &= d_{i+1} + l_i^2 d_i, \quad 1 \le i \le n-1 \end{aligned}.$$

Combining the equations and solving d_{i+1}, we get

$$d_1 = \alpha_1 - \sigma,$$
$$d_{i+1} = (\alpha_{i+1} - \sigma) - \frac{\beta_i^2}{d_i}, \qquad 1 \le i \le n-1$$

Thus, we get a recurrence formula. Though it is a recurrence formula, it is not linear. And, to compute a nonlinear recurrence in parallel is not easy. In order to calculate $d_1, d_2, \cdots, d_n$ in parallel, the equations

$$d_1 = \alpha_1 - \sigma,$$
$$d_{i+1} = (\alpha_{i+1} - \sigma) - \frac{\beta_i^2}{d_i}, \qquad 1 \le i \le n-1$$

can be transformed into a second order linear recurrence by letting

$$d_1 = \frac{\alpha_1 - \sigma}{1} = \frac{g_1}{g_0}$$
$$\begin{aligned} d_{i+1} &= (\alpha_{i+1} - \sigma) - \frac{\beta_i^2}{d_i} \\ &= \frac{(\alpha_{i+1} - \sigma) g_i - \beta_i^2 g_{i-1}}{g_i} \\ &= \frac{g_{i+1}}{g_i}, \qquad 1 \le i \le n-1. \end{aligned}$$

Then we get a second order linear recurrence

$$g_0 = 1,$$
$$g_1 = \alpha_1 - \sigma,$$
$$g_{i+1} = (\alpha_{i+1} - \sigma) g_i - \beta_i^2 g_{i-1}, \qquad 1 \le i \le n-1.$$

And we can compute

$$d_{i+1} = \frac{g_{i+1}}{gi}, \qquad 0 \le i \le n-1.$$

There are many parallel algorithm to solve the second order linear recurrence g_i, $1 \le i \le n-1$, [7]. For different values of σ, the number of positive d_i, $1 \le i \le n$, is equal to the number of eigenvalues above the sample point σ. Repeated applications of this procedure will separate the eigenvalue spectrum into small sub-intervals of size ϵ which contain one or more eigenvalues of matrix A.

3 The Real Symmetric Circulant Tridiagonal Matrix

In this section, we shall propose an algorithm, based on the bisection method, to solve the eigenvalue problem on the real symmetric circulant tridiagonal matrix. The form of a symmetric circulant tridiagonal matrix A is

$$A = \begin{bmatrix} \alpha_1 & \beta_1 & & & c \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{n-2} & \alpha_{n-1} & \beta_{n-1} \\ c & & & \beta_{n-1} & \alpha_n \end{bmatrix}.$$

For any given $\sigma \in R$, we shall determine the inertia of $A - \sigma I$. And $A - \sigma I$ can be decomposed into $L_\sigma D_\sigma L_\sigma^T$. Our factorization of $A - \sigma I$ is

$$A - \sigma I = \begin{bmatrix} \alpha_1 - \sigma & \beta_1 & & & c \\ \beta_1 & \alpha_2 - \sigma & \beta_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{n-2} & \alpha_{n-1} - \sigma & \beta_{n-1} \\ c & & & \beta_{n-1} & \alpha_n - \sigma \end{bmatrix} =$$

$$\begin{bmatrix} 1 & & & & & \\ l_1 & 1 & & & & \\ & l_2 & 1 & & & \\ & & \ddots & \ddots & & \\ x_1 & \cdots & x_{n-2} & l_{n-1} & 1 \end{bmatrix} \begin{bmatrix} d_1 & & & & \\ & d_2 & & & \\ & & \ddots & & \\ & & & d_{n-1} & \\ & & & & d_n \end{bmatrix}$$

$$\begin{bmatrix} 1 & l_1 & & & x_1 \\ & 1 & l_2 & & \vdots \\ & & \ddots & \ddots & x_{n-2} \\ & & & 1 & l_{n-1} \\ & & & & 1 \end{bmatrix}.$$

Then we have

$$\beta_i = l_i d_i, \qquad 1 \le i \le n-2$$
$$\beta_{n-1} = x_{n-2} d_{n-2} l_{n-2} + d_{n-1} l_{n-1},$$
$$x_1 d_1 = c,$$
$$x_i d_i l_i + x_{i+1} d_{i+1} = 0, \quad 1 \le i \le n-3$$

and

$$\alpha_1 - \sigma = d_1,$$
$$\alpha_{i+1} - \sigma = l_i^2 d_i + d_{i+1}, \qquad 1 \le i \le n-2$$
$$\begin{aligned} \alpha_n - \sigma = (&x_1^2 d_1 + x_2^2 d_2 + \cdots + x_{n-2}^2 d_{n-2} \\ &+ l_{n-1}^2 d_{n-1}) + d_n. \end{aligned}$$

Solving l_i, x_i, and d_i from the above equations, we obtain

$$l_i = \frac{\beta_i}{d_i}, \qquad 1 \le i \le n-2$$
$$l_{n-1} = \frac{(\beta_{n-1} - \beta_{n-2} x_{n-2})}{d_{n-1}},$$
$$x_1 = \frac{c}{d_1},$$
$$x_{i+1} = -\left(\frac{d_i l_i}{d_{i+1}}\right) x_i, \qquad 1 \le i \le n-3$$

and

$$d_1 = \alpha_1 - \sigma$$
$$d_{i+1} = (\alpha_{i+1} - \sigma) - l_i^2 d_i, \qquad 1 \le i \le n-2$$
$$\begin{aligned} d_n = (\alpha_n - \sigma) - (&x_1^2 d_1 + x_2^2 d_2 + \cdots \\ &+ x_{n-2}^2 d_{n-2} + l_{n-1}^2 d_{n-1}) \end{aligned}$$

Combing l_i and x_i, we have

$$x_1 = \frac{c}{d_1}$$
$$x_{i+1} = -(\frac{\beta_i}{d_{i+1}})x_i, \quad 1 \leq i \leq n-3$$

and combing l_i and d_i we get

$$d_1 = \alpha_1 - \sigma$$
$$d_{i+1} = (\alpha_{i+1} - \sigma) - (\frac{\beta_i^2}{d_i}) \qquad i = 1, 2, \cdots, n-2.$$
$$d_n = (\alpha_n - \sigma) - (x_1^2 d_1 + x_2^2 d_2 + \cdots + x_{n-2}^2 d_{n-2} + \frac{(\beta_{n-1} - \beta_{n-2} x_{n-2})^2}{d_{n-1}}).$$

As the work in the previous section, we can calculate d_i, $1 \leq i \leq n-1$, by transforming d_i into a second order linear recurrence sequence g_i. And then we compute the linear recurrence sequence x_i, $2 \leq i \leq n-2$ to get d_n.

4 The Symmetric Quindiagonal Matrix

In this section, we shall present how the Sylvester's law of inertia can be applied to solve the eigenvalue problem on the *real symmetric quindiagonal matrix*. Consider the $n \times n$ real symmetric quindiagonal matrix

$$A = \begin{bmatrix} \alpha_1 & \beta_1 & \gamma_1 & & & & \\ \beta_1 & \alpha_2 & \beta_2 & \gamma_2 & & & \\ \gamma_1 & \beta_2 & \alpha_3 & \beta_3 & \gamma_3 & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & \gamma_{n-4} & \beta_{n-3} & \alpha_{n-2} & \beta_{n-2} & \gamma_{n-2} \\ & & & \gamma_{n-3} & \beta_{n-2} & \alpha_{n-1} & \beta_{n-1} \\ & & & & \gamma_{n-2} & \beta_{n-1} & \alpha_n \end{bmatrix}.$$

Evans [5] had derived a recursive formula from the determinantal equation

$$f_k(\lambda) = det(A_k - \lambda I),\ 1 \leq k \leq n,$$

where A_k is the leading submatrix of size $k \times k$ containing the entry $(1,1)$ of A.

The recurrence sequence is

$$f_0(\lambda) = 1$$
$$f_1(\lambda) = (\alpha_1 - \lambda) f_0(\lambda)$$
$$f_2(\lambda) = (\alpha_2 - \lambda) f_1(\lambda) - \beta_1^2 f_0(\lambda)$$
$$f_3(\lambda) = (\alpha_3 - \lambda) f_2(\lambda) - \beta_2^2 f_1(\lambda) - \gamma_1^2(\alpha_2 - \lambda) + 2\beta_1\beta_2\gamma_1$$
$$\vdots$$
$$f_k(\lambda) = (\alpha_k - \lambda) f_{k-1}(\lambda) - \beta_{k-1}^2 f_{k-2}(\lambda) - \gamma_{k-2}^2\{(\alpha_{k-1} - \lambda) f_{k-3}(\lambda) - \gamma_{k-3}^2 f_{k-4}(\lambda)\}$$
$$+2\sum_{j=1}^{k-2}(-1)^{j+1}\beta_{k-1}\beta_{k-j-1}\left[\prod_{r=k-j+1}^{k}\gamma_{r-2}\right] f_{k-j-2}(\lambda),$$
$$4 \leq k \leq n.$$

Newton's iterative method can be used to find the zeros of $f_n(\lambda)$, a polynomial of degree n. Then these zeros are the eigenvalues of A. The Newton iteration has the form

$$\lambda_{k+1} = \lambda_k - [f_n(\lambda_k)/f_n'(\lambda_k)],$$

where λ_0 is an initial estimate.

Graham [1] calculated the eigenvalues of a real symmetric quindiagonal matrix by using cubic polynomial interpolation, as well as first and second order Newton iteration method to solve the scaled recursive formula. The scaled recursive formula, based on the Evans' [5] recursive formula, uses a scaling which maintains the correct relative magnitudes in the iterative process.

Our algorithm for solving the eigenvalue problem of the real symmetric quindiagonal matrix is to partition the quindiagonal matrix into the symmetric block tridiagonal matrix form. Then we apply the Sylvester's law of inertia to solve the symmetric block tridiagonal matrix.

Consider the $m \times m$ symmetric block tridiagonal matrix

$$A = \begin{bmatrix} A_1 & B_1^T & & & \\ B_1 & A_2 & B_2^T & & \\ & \ddots & \ddots & \ddots & \\ & & B_{m-2} & A_{m-1} & B_{m-1}^T \\ & & & B_{m-1} & A_m \end{bmatrix},$$

where A_i and B_i, $1 \leq i \leq m$, denote $p \times p$ submatrices. Because A is symmetric, for any given $\sigma \in R$, we can determine the inertia of $A - \sigma I$ by decomposing it into $L_\sigma D_\sigma L_\sigma^T$. And the decomposition can be expressed as

$$A - \sigma I = \begin{bmatrix} A_1 - \sigma I & B_1^T & & & \\ B_1 & A_2 - \sigma I & B_2^T & & \\ & \ddots & \ddots & \ddots & \\ & & B_{m-2} & A_{m-1} - \sigma I & B_{m-1}^T \\ & & & B_{m-1} & A_m - \sigma I \end{bmatrix} =$$
$$\begin{bmatrix} I & & & & \\ L_1 & I & & & \\ & \ddots & \ddots & & \\ & & L_{m-2} & I & \\ & & & L_{m-1} & I \end{bmatrix} \begin{bmatrix} D_1 & & & & \\ & D_2 & & & \\ & & \ddots & & \\ & & & D_{m-1} & \\ & & & & D_m \end{bmatrix} \begin{bmatrix} I & L_1^T & & & \\ & I & L_2^T & & \\ & & \ddots & \ddots & \\ & & & I & L_{m-1}^T \\ & & & & I \end{bmatrix},$$

where D_k, $1 \leq k \leq m$, is a submatrix of D_σ, and L_k, $1 \leq k \leq m-1$, is a submatrix of L_σ. Note that D_k is also symmetric since $A - \sigma I$ is symmetric.

By multiplying the right side, we get

$$A_1 - \sigma I = D_1$$
$$B_k = L_k D_k$$
$$(A_{k+1} - \sigma I) = D_{k+1} + L_k D_k L_k^T,$$
$$1 \leq k \leq m.$$

Combining $L_k = B_k D_k^{-1}$ and $(A_{k+1} - \sigma I)$, we get

$$D_1 = A_1 - \sigma I$$
$$D_{k+1} = (A_{k+1} - \sigma I) - B_k D_k^{-1} B_k^T,$$

$$1 \le k \le m-1.$$

Submatrix D_{k+1} can be computed from D_k. The inertia $\pi(A-\sigma I)$ is the same as $\pi(D_\sigma)$. The matrix D_σ is a block diagonal matrix, so the eigenvalues of D_σ are the union of the submatrices of D_k, $1 \le k \le m-1$. The eigenvalues of each D_k can be solved by any eigenvalue solver independently on a multiprocessor. Then the number $\pi(D_\sigma)$ can be computed from the eigenvalues of D_k.

Next, we apply the block matrix notation on the symmetric quindiagonal matrix. The 2×2 partition of the symmetric quindiagonal matrix $A \in R^{n \times n}$ can be represented in the form

$$A_k = \begin{bmatrix} \alpha_{2k-1} & \beta_{2k-1} \\ \beta_{2k-1} & \alpha_{2k} \end{bmatrix},$$

and

$$B_k = \begin{bmatrix} \gamma_{2k-1} & \beta_{2k} \\ 0 & \gamma_{2k} \end{bmatrix},$$

where $1 \le k \le m$ and the size of matrix A is $n = 2m$.

For a given $\sigma \in R$, we can compute $\pi(A-\sigma I)$. By the block matrix decomposition $A - \sigma I = L_\sigma D_\sigma L_\sigma$, we can get submatrix D_{k+1} of D_σ by the formula

$$D_{k+1} = (A_{k+1} - \sigma I) - B_k D_k^{-1} B_k^T.$$

Assume

$$D_k = \begin{bmatrix} d_{2k-1} & c_{2k-1} \\ c_{2k-1} & d_{2k} \end{bmatrix},$$

and

$$D_{k+1} = \begin{bmatrix} d_{2(k+1)-1} & c_{2(k+1)-1} \\ c_{2(k+1)-1} & d_{2(k+1)} \end{bmatrix}.$$

Then we can compute D_{k+1} from

$$\begin{bmatrix} \alpha_{2(k+1)-1} - \sigma & \beta_{2(k+1)-1} \\ \beta_{2(k+1)-1} & \alpha_{2(k+1)} - \sigma \end{bmatrix} - \begin{bmatrix} \gamma_{2k-1} & \beta_{2k} \\ 0 & \gamma_{2k} \end{bmatrix} \begin{bmatrix} d_{2k-1} & c_{2k-1} \\ c_{2k-1} & d_{2k} \end{bmatrix}^{-1} \begin{bmatrix} \gamma_{2k-1} & 0 \\ \beta_{2k} & \gamma_{2k} \end{bmatrix}.$$

From the above matrix computation, the matrix entries of D_{k+1} can be calculated as

$$d_{2(k+1)-1} = (\alpha_{2(k+1)-1} - \sigma) - \frac{1}{u}(d_{2k}\gamma_{2k-1}^2 - 2c_{2k-1}\gamma_{2k-1}\beta_{2k} + d_{2k-1}\beta_{2k}^2)$$

$$d_{2(k+1)} = (\alpha_{2(k+1)} - \sigma) - \frac{1}{u}(d_{2k-1}\gamma_{2k}^2)$$

$$c_{2(k+1)-1} = \beta_{2(k+1)-1} - \frac{1}{u}(c_{2k-1}\gamma_{2k-1}\gamma_{2k} - d_{2k-1}\gamma_{2k}\beta_{2k-1}), \qquad 1 \le k \le m,$$

where $u = d_{2k-1}d_{2k} - c_{2k-1}^2$.

The matrix inertia $\pi(A-\sigma I)$ for a given $\sigma \in R$ can be computed after the eigenvalues of those 2×2 submatrices are obtained. It is easier than the eigenvalues calculation on the original quindiagonal matrix A. And, from the block matrix notation, the formula

$$D_{k+1} = (a_{k+1} - \sigma I) - B_k D_k^{-1} B_k^T$$

is simpler than the recursive formula based on the determinant, proposed by Evans and Yousif [5].

5 Time Complexity

In this section, we will analyze the time complexities of our algorithms for solving the eigenvalues of the symmetric circulant tridiagonal and symmetric quindiagonal matrices. Because our algorithms are based on the Sylvester's law of inertia, finding the inertia of the given matrix cost most of the time in the algorithm. For given $\sigma \in R$, we analyze the time complexity of computing the inertia of the matrix corresponding to σ.

In the symmetric tridiagonal matrix, finding the inertia of the matrix is the same as computing the second order linear recurrence sequence g_i, $1 \le i \le n$. In parallel computing, the time complexity is $O(\log n)$ by using $O(n/\log n)$ processors.

Computing the inertia of the symmetric circulant tridiagonal matrix with a real number $\sigma \in R$ can be divided into the following three steps:

Step1. Computing d_i, $1 \le i \le n-2$, can be transformed into a second order linear recurrence sequence. Its time complexity is $O(\log n)$ by using $O(n/\log n) processors$.

Step2. We also compute a linear recurrence sequence x_i, $1 \le i \le n-3$. Its time complexity is $O(\log n)$ by using $O(n/\log n)$ processors.

Step3. We use x_i and d_i to compute d_n, and the time complexity of this step is also $O(\log n)$ by using $O(n/\log n)$ processors.

Then we get that the time complexity is $O(\log n)$ and the number of processors is $O(n/\log n)$ for computing the inertia of a symmetric circulant tridiagonal matrix for given $\sigma \in R$.

Next, for the symmetric quindiagonal matrix, we first compute the submatrices D_k, $1 \le k \le m$, which must be done sequentially. The time complexity is $O(n)$ to get the submatrices, because $m = \frac{n}{2}$, and then computing eigenvalues of the m submatrices parallelly, it costs constant time to get the eigenvalues of each submatrices D_k. Thus, the time complexity is $O(n)$ with $O(n)$ processors to get the inertia of the symmetric quindiagonal matrix.

6 Conclusion

Computing eigenvalues of matrices is important for many scientific areas. Much study has been done on the eigenvalue problem, including the parallel algorithms for this problem. However, most of these papers are to study the eigenvalues of the symmetric tridiagonal matrix. Some of the algorithms, such as the QR method, can compute all the eigenvalues of a matrix. However, it is time consuming to do this. And sometimes we are not interested in all eigenvalues of a matrix. The bisection method is a simple and inexpensive procedure for calculating some specified eigenvalues of a matrix.

Our algorithms are based on the bisection method and the Sylvester's law of inertia to solve some specified eigenvalues of symmetric circulant tridiagonal and symmetric quindiagonal matrices. For the symmetric circulant tridiagonal matrix, we also propose a parallel algorithm. Same as the symmetric tridiagonal matrix, its time complexity is $O(logn)$ by using $O(n)$ processors. For the symmetric quindiagonal matrix, we present a simpler algorithm than the determination equation given by Evans [5].

References

[1] G. Baker, "Accelerated bisection techniques for tri- and quindiagonal matrices," *International Journal for Numerical Methods in Engineering*, Vol. 35, pp. 203–218, 1992.

[2] R. H. Bartels, J. C. Beatty, and B. A. Barsky, *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling*. San Mateo, CA: Morgan Kaufman, 1987.

[3] K. L. Chung, Y. H. Tsai, and W. Yang, "A parallel solver for circulant block tridiagonal systems," *Computers & Mathematics with Applications*, Vol. 29, No. 1, pp. 109–113, 1995.

[4] J. Cuppen, "A divide and conquer method for the symmetric tridiagonal eigenproblem," *Numeriche Mathematik*, Vol. 36, pp. 177–195, 1981.

[5] D. J. Evans, "A recursive algorithm for determining the eigenvalues of a quindiagonal matrix," *The Computer Journal*, Vol. 18, pp. 70–73, 1975.

[6] G. H. Golub and C. F. V. Loan, *Matrix Computations*. Baltimore and London: The Johns Hopkins University, 1983.

[7] S. Gurak, "Pseudoprimes for higher-order linear recurrence sequences," *Mathematics of Computation*, Vol. 55, No. 192, pp. 783–813, 1990.

[8] B. O. Han and B. Codenotti, "A fast and stable parallel QR algorithm for symmetric tridiagonal matrices," *Linear Algebra and Its Applications*, Vol. 220, pp. 63–95, 1995.

[9] L. Kaufman, "An algorithm for the banded symmetric generalized matrix eigenvalue problem," *SIAM Journal on Matrix Analysis and Applications*, Vol. 14, No. 2, pp. 372–389, 1993.

[10] M. Lu and X. Qiao, "Applying parallel computer systems to solve symmetric tridiagonal eigenvalue problems," *Parallel Computing*, Vol. 18, pp. 1301–1315, 1992.

[11] O. Rojo, "A new method for solving symmetric circulant tridiagonal systems of linear equations," *Computers & Mathematics with Applications*, Vol. 20, No. 12, pp. 61–67, 1990.

[12] H. Sagan, *Boundary and Eigenvalue Problems in Mathematical Physics*. New York: Dover Publications,Inc, 1989.

[13] P. N. Swarztrauber, "A parallel algorithm for computing the eigenvalues of a symmetric tridiagonal matrix," *Mathematics of Computation*, Vol. 60, No. 202, pp. 651–668, 1993.

[14] D. B. Szyld, "Criteria for combining inverse and Rayleigh quotient iteration," *SIAM Journal on Numerical Analysis*, Vol. 25, No. 6, pp. 1369–1375, 1988.

[15] C. Trefftz, C. C. Huang, P. K. Mckinley, T. Y. Li, and Z. Zeng, "A scalable eigenvalue solver for symmetric tridiagonal matrices," *Parallel Computing*, Vol. 21, pp. 1213–1240, 1995.