<table>
<thead>
<tr>
<th>Stage</th>
<th>Bisect</th>
<th>p=2</th>
<th>p=4</th>
<th>p=8</th>
<th>p=16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Init.</td>
<td>95.36</td>
<td>50.98</td>
<td>25.53</td>
<td>13.75</td>
<td>6.89</td>
</tr>
<tr>
<td>s=1</td>
<td></td>
<td>33.25</td>
<td>17.12</td>
<td>9.65</td>
<td>5.25</td>
</tr>
<tr>
<td>s=2</td>
<td></td>
<td></td>
<td>32.55</td>
<td>17.43</td>
<td>8.95</td>
</tr>
<tr>
<td>s=3</td>
<td></td>
<td></td>
<td></td>
<td>34.01</td>
<td>18.27</td>
</tr>
<tr>
<td>s=4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>37.48</td>
</tr>
<tr>
<td>Total</td>
<td>95.36</td>
<td>84.23</td>
<td>75.20</td>
<td>74.84</td>
<td>76.84</td>
</tr>
</tbody>
</table>

Table 1

Computing time in seconds, for N = 8191 and k = 457.

REFERENCES

Stage $s=1,\ldots,t$: We compute in parallel the eigenvalues of

$$T_i^{(s)}, \quad i = 1,\ldots,l, \quad l = q/2^s,$$

in the interval $\mathcal{I}$ in the following way. We first merge the set of eigenvalues of $T_{2i-1}^{(s-1)}$ with those of $T_{2i}^{(s-1)}$ to get,

$$\Gamma^{(s)} = \{ \lambda_i^{(s)} \} = -\infty < \lambda_1^{(s)} \leq \cdots \leq \lambda_{m_i}^{(s)} < \infty = \lambda_{m_i+1}^{(s)} 
$$

where $m_i = n_{(2i-1),(s-1)} + n_{(2i),(s-1)}$. Then we subdivide these $p/2^s$ sets of intervals between the processors letting each processor have the same number of eigenvalues to compute. This can be done using partial summation, see for example Bar-On and Vishkin[4]. Each processor then computes its set of eigenvalues using the QR algorithm together with bisection. The middle eigenvalue is first computed using bisection and then the QR algorithm is used, with an initial shift of the computed middle eigenvalue, and further shifts according to the interlacing bounds. The QR algorithm will then converge fast to the eigenvalues surrounding the middle one. We denote the number of eigenvalues in each submatrix $T_i$ by $m_i - 1 \leq n_i \leq m_i + 1$.

**Complexity:** Let there be $k \geq p$ eigenvalues of $T$ in the interval $\mathcal{I}$. Then, at the Initialization Stage we compute at most $k + p - 1$ eigenvalues, for a total of $O(kn)$ time. In Stage $s=1,\ldots,t$, we compute at most $k + p - 1$ eigenvalues using $p$ processors for a total of $O(kn2^s/p)$ time. Hence, the total complexity of the algorithm is $O(kN/p)$ and the algorithm is efficient.

**Numerical examples:** We have implemented a preliminary version of the algorithm by simulation on the DEC AXP 3000-500 machine, under UNIX OSF/1 operating system. We have compared the running time of the simulated parallel algorithm with that of the corresponding routine Bisect in EISPACK. The results for matrix $T = Tridiag(-1,2,-1)$ with $p \leq 16$ processors, and $N = 2^{13} - 1$ are given in Table 1. As can be seen from this example, the last stage in the simulated algorithm is about three times faster than the corresponding bisection algorithm. This is due to the fact that we can use the interlacing properties to accelerate the running time of the QR algorithm. The whole simulation is also faster than the sequential method, so we may expect the parallel implementation to be very efficient.

5. **Conclusion.** We have presented new theoretical results relating the eigenvalues of a tridiagonal symmetric matrix to those of its leading and trailing sub matrices. We have then developed a new divide and conquer algorithm for computing the whole spectrum of the matrix using the secular equation and a partial set of the spectrum using Bisection and the QR algorithm. Some preliminary numerical tests have demonstrated the efficiency of these methods, and we are working on further optimizing these codes for computing the whole and partial sets of the spectrum.
Hence, we have,

\begin{align*}
(57) \\
    y &= b(\lambda I - T_1)^{-1}e_m = bQ_1(\lambda I - D_1)^{-1}q^{m(1)}, \\
    z &= c(\lambda I - T_2)^{-1}e_1 = cQ_2(\lambda I - D_2)^{-1}q^{1(2)}.
\end{align*}

Let, \( v \) be the corresponding normalized eigenvector, then

\begin{align*}
(58) \\
    v_1 &= y_1 / \| x \|, \quad v_{n_s-1} = z_m / \| x \|,
\end{align*}

with,

\begin{align*}
(59) \\
    y_1 &= e_1^T y = b(q^{1(1)})^T (\lambda I - D_1)^{-1}q^{m(1)}, \\
    z_m &= e_m^T z = b(q^{m(2)})^T (\lambda I - D_2)^{-1}q^{1(2)},
\end{align*}

and,

\begin{align*}
(61) \\
    \| y \|^2 &= b^2(q^{m(1)})^T (\lambda I - D_1)^{-2}q^{m(1)}, \\
    (62) \\
    \| z \|^2 &= c^2(q^{1(2)})^T (\lambda I - D_2)^{-2}q^{1(2)},
\end{align*}

and finally, \( \| x \| = (\| y \|^2 + 1 + \| z \|^2)^{1/2} \).

**Complexity:** The complexity of the Initialization Stage is \( O(n^2) \), and that of Stage \( s=1 \ldots t \), is \( O((2^s n)^2 (N/(2^s n))) = O(N^2/2^{-s}) \). Hence, we can compute the complete set of eigenvalues in \( O(N^2) \) time.

### 4.3. A parallel divide and conquer algorithm for computing a partial set of the eigenvalues

We develop in this subsection a parallel algorithm for computing the whole set of eigenvalues in a given interval. Here we may assume that the matrix is very large so that the computation of the whole spectrum is not feasible. We then use the interlacing properties of the eigenvalues to subdivide the interval into subintervals each containing the same number of eigenvalues, and we let each processor work on a different subinterval in parallel. In what follows we assume a parallel architecture with \( p \) independent processors, each having its own local memory, and a reasonably fast interconnection network that allows for example merging of small sets to be done efficiently. We further assume for simplicity that \( p = q \) in the notations of (32), and that a copy of the tridiagonal matrix can reside in the memory of each processor (which is usually the case unless \( T \) is very huge, see[3][2]). Finally, we denote the endpoints of the interval by \( I = [x_l, x_r] \). The whole process is described in the following.

**Initialization:** We compute in parallel the eigenvalues of

\begin{equation}
(63) \\
    T_i^{(0)}, \quad i = 1, \ldots, p,
\end{equation}

in the interval \( I \) using Bisection as in[12]. We denote the number of such eigenvalues in each submatrix by \( n_{i_0} \).
Then,

$$y = (\lambda I - D_1)^{-1} f, \quad z = (\lambda I - D_2)^{-1} g,$$

and $f^T y + a + g^T z = \lambda$. Hence,

$$0 = (\lambda - a) - f^T (\lambda I - D_1)^{-1} f - g^T (\lambda I - D_2)^{-1} g,$$

which is equivalent to (47). \[\square\]

**Corollary 4.2.** The eigenvalues $\lambda \notin \lambda^{(1)} \cup \lambda^{(2)}$ of $T$, are the roots of the secular equation $S(\lambda)$ in (47). This function is monotone increasing in each subinterval, and we can use fast methods to compute its roots; see for example Dongarra and Sorensen[7].

We now proceed to consider the computation of the corresponding extreme elements of each eigenvector.

**Case** $\lambda \in \lambda^{(1)} \cap \lambda^{(2)}$ : Let $y, z \in \mathbb{R}^m$ be the respective unit eigenvectors,

$$T_1 y = \lambda y, \quad T_2 z = \lambda z,$$

where we assume w.l.o.g. that $|by_m| \leq |cz_1|$. We then choose as eigenvector $x = (y, 0, \omega z)^T$, where $\omega$ is yet to be chosen. Hence,

$$Tx = \begin{pmatrix} T_1 y \\ by_m + \omega cz_1 \\ T_2 \omega z \end{pmatrix} = \begin{pmatrix} \lambda y \\ by_m + \omega cz_1 \\ \lambda \omega z \end{pmatrix},$$

and we take,

$$\omega = -\frac{by_m}{cz_1},$$

which is possible since $cz_1 \neq 0$. The extreme components of the normalized vector are then,

$$v_1 = y_1 / \| x \|, \quad v_{n-1} = \omega z_m / \| x \|,$$

with,

$$\| x \| = (\| y \|^2 + \omega^2 \| z \|^2)^{1/2} = (1 + \omega^2)^{1/2}.$$

**Case** $\lambda \notin \lambda^{(1)} \cup \lambda^{(2)}$ : Let $(\lambda, x)$ be a corresponding pair for $T$, with $x = (y, 1, z)^T$. Then,

$$Tx = \begin{pmatrix} T_1 y + be_m \\ by_m + a + cz_1 \\ ce_1 + T_2 z \end{pmatrix} = \lambda \begin{pmatrix} y \\ 1 \\ z \end{pmatrix}.$$
Initialization: Consider the spectral decomposition of $T_i^{(0)}$,

$$D_i^{(0)} = (Q_i^{(0)})^T T_i^{(0)} Q_i^{(0)}, \quad i = 1, \ldots, q.$$  

Then, we compute with each eigenvalues, the corresponding extreme components of its eigenvector,

$$q^{(i)} = (c_i Q_i^{(0)})^T, \quad j = 1, n - 1.$$

Stage $s=1, \ldots, t$: We simplify the notation and write

$$T = \begin{pmatrix} T_1 & b & a & c \\ b & c & T_2\end{pmatrix}, \quad T \in \mathcal{M}(n, -1)$$

for $T_i^s$, and

$$\lambda^{(k)} = \{\lambda_0^{(k)} = -\infty < \lambda_1^{(k)} < \cdots < \lambda_m^{(k)} < \infty = \lambda_{m+1}^{(k)}\}, \quad k = 1, 2$$

for the eigenvalues of $T_1 = T_2^{(s-1)}$ and $T_2 = T_2^{(s-1)}$. We similarly denote the corresponding extreme components of each eigenvector by,

$$q^{(k)} = Q_k^T e_j, \quad j = 1, m, \quad k = 1, 2.$$

We then proceed to compute the set of eigenvalues of $T$, together with the corresponding set of extreme vectors using the decomposition:

$$D = \hat{Q}^T T \hat{Q} = \begin{pmatrix} D_1 & f^T & a & g^T \\ f & a & g & D_2\end{pmatrix}, \quad \hat{Q} = \begin{pmatrix} Q_1 & 1 \\ & & Q_2\end{pmatrix},$$

where,

$$f = b Q_1 e_m = b q^{(1)}, \quad g = c Q_2 e_1 = c q^{(1)}.$$

We note that for $\lambda \in \lambda^{(1)} \cap \lambda^{(2)}$ we have already an eigenvalue of $T$, and that numerically, this is also the case when there are two relatively close eigenvalues in $\lambda^{(1)} \cup \lambda^{(2)}$. Similarly, this is the case when some of components of $f$ or $g$ are relatively small.

**Lemma 4.1.** For $\lambda \notin \lambda^{(1)} \cup \lambda^{(2)}$ we have,

$$S(\lambda) = (\lambda - a) + \sum_{i=1}^{m} \frac{f_i^2}{(\lambda_i^{(1)} - \lambda)} + \sum_{i=1}^{m} \frac{g_i^2}{(\lambda_i^{(2)} - \lambda)} = 0.$$  

*Proof.* Let $(\lambda, x)$ be a pair of a corresponding eigenvalue and eigenvector of $D$ in (45), say,

$$D x = \begin{pmatrix} D_1 & f^T & a & g^T \\ f & a & g & D_2\end{pmatrix} \begin{pmatrix} y \\ 1 \\ z\end{pmatrix} = \lambda \begin{pmatrix} y \\ 1 \\ z\end{pmatrix}.$$
4.1. The basic algorithm. We begin with a general description of the basic algorithm.

Initialization: Compute all the eigenvalues of,

\[ T_i^{(0)}, \quad i = 1, \ldots, q, \]

using any sequential method such as Bisection[12].

Stage s=1, \ldots, t: Compute all the eigenvalues of

\[ T_i^{(s)}, \quad i = 1, \ldots, l, \quad l = q/2^s, \]

in the following way. Let,

\[ T_i^{(s)} = \begin{pmatrix} T_{2i-1}^{(s-1)} & \gamma_{2i-1}^{(s-1)} \\ \gamma_{2i-1}^{(s-1)} & \lambda_{2i-1}^{(s-1)} \end{pmatrix}, \]

and denote its set of eigenvalues by,

\[ \lambda_i^{(s)} = \{ \lambda_0^{(s)} = -\infty < \lambda_1^{(s)} < \cdots < \lambda_{n_s-1}^{(s)} < \infty = \lambda_{n_s}^{(s)} \}, \quad n_s = 2^s n. \]

Let, \((T_i^{(s)})_{(n_s-1)}\) be obtained from \(T_i^{(s)}\) by deleting the middle row and column, and let us denote its set of eigenvalues by,

\[ \Gamma_i^{(s)} = \{ \gamma_0^{(s)} = -\infty < \gamma_1^{(s)} \leq \cdots \leq \gamma_{n_s-2}^{(s)} < \infty = \gamma_{n_s-1}^{(s)} \}. \]

Then we conclude from the foregoing that

\[ \lambda_j^{(s)} \in (\gamma_{j-1}^{(s)}, \gamma_j^{(s)}), \quad j = 1, \ldots, n_s - 1, \]

which is already a crude and an isolating estimate. The computation can then proceed using some root finding technique. These idea will be further elaborated in the following subsections.

4.2. Computing the whole spectrum of the matrix. We develop in this subsection a root finding technique based on the secular equation. This method relies on the computation of the extreme components of the eigenvectors and is appropriate when we want to compute the whole spectrum. However, when only a partial set of the eigenvalues is needed, the method is inefficient since it requires \(O(N^2)\) time. A method for locating the eigenvalues based on the interlacing properties alone is then developed in the next subsection.
Lemma 3.6. For $1 \leq i \leq j \leq N$, we have

\[(27) \quad \gamma_{i-1}^{(k_1)} \leq \gamma_j^{(k_2)}, \quad 1 \leq k_1, k_2 \leq r.\]

Proof. Using Theorem (3.1) we obtain,

\[(28) \quad \lambda_i \in (\gamma_{i-1}^{(k_1)}, \gamma_i^{(k_1)}), \quad \lambda_j \in (\gamma_j^{(k_2)}, \gamma_j^{(k_2)}).\]

Suppose, $\gamma_j^{(k_2)} < \gamma_i^{(k_1)}$, then $\lambda_i > \lambda_j$, a contradiction since $i \leq j$. \(\square\)

Proof. Using Theorem (3.1) we obtain,

\[(29) \quad \lambda_i \in \bigcap_{k=1}^r (\gamma_i^{(k)}), \quad (\max_{1 \leq k \leq r} \gamma_i^{(k)}, \min_{1 \leq k \leq r} \gamma_i^{(k)}).\]

Since by the Lemma,

\[(30) \quad \gamma_i^{(k_1)} \leq \gamma_i^{(k_2)}, \quad 1 \leq k_1, k_2 \leq r, \quad 1 \leq i \leq (N-1),\]

we may assume w.l.g. that

\[(31) \quad \gamma_i^{(k)}, \quad k = 1, \ldots, r \quad \text{for} \quad i = 1, \ldots, (N-1),\]

appears consecutively in $E$. \(\square\)

4. Divide and Conquer. We develop in this section a new divide and conquer algorithm for computing the eigenvalues of the tridiagonal matrix $T \in \mathcal{M}(N)$ based on the interlacing properties developed in the previous section. We will assume here for simplicity that $T$ is unreduced, and that $N = nq - 1$, $q = 2^l$. The generalization is then straightforward. We further use block representation,

\[(32) \quad T^{(s)} = \begin{pmatrix} T_1^{(s)} & b_1^{(s)} & a_1^{(s)} & c_1^{(s)} & \cdots & b_1^{(s)} \\ b_1^{(s)} & a_1^{(s)} & c_1^{(s)} & \cdots & b_1^{(s)} \\ & \ddots & \ddots & \ddots & \ddots & \ddots \\ & & \ddots & \ddots & b_{l-1}^{(s)} & a_{l-1}^{(s)} & c_{l-1}^{(s)} \\ & & & b_{l-1}^{(s)} & a_{l-1}^{(s)} & c_{l-1}^{(s)} & T_1^{(s)} \end{pmatrix}, \quad T_i^{(s)} \in \mathcal{M}(n_s, -1), \quad \begin{array}{l} i = 1, \ldots, l, \\ l = q/2^s, \quad n_s = 2^s n, \\ s = 0, \ldots, l, \end{array}\]

where in relation to (2) we have

\[(33) \quad b_i^{(s)} = b_{m_s-1}, \quad a_i^{(s)} = a_{m_s}, \quad c_i^{(s)} = b_{m_s}.\]
We finally prove Theorem (3.1):

Proof. Given a real number $x$ which is not an eigenvalue of $T_k$ nor of $H_{k+2}$, we let,

$$F^T = \begin{pmatrix} I_k & b_k(e_k^{(k)})^T(T_k - xI)^{-1} & -b_{k+1}(e_{k+1}^{(m)})^T(H_{k+2} - xI)^{-1} \\ I_m & 1 & -xI \end{pmatrix},$$

so that,

$$F^T(T - xI)F = \begin{pmatrix} T_k - xI_k & a_{k+1}(x) \\ a_{k+1}(x) & H_{k+2} - xI_m \end{pmatrix}.$$

Hence, by the Sylvester Inertia theorem[10],

$$\pi(T_k - xI) + \pi(H_{k+2} - xI) \leq \pi(T - xI) \leq \pi(T_k - xI) + \pi(H_{k+2} - xI) + 1,$$

where $\pi(A)$ denotes the number of negative eigenvalues of $A$. Therefore, the number of eigenvalues inside a non-redundant interval is bounded by,

$$\pi(T - xI) - \pi(T - yI) \leq 1,$$

for some $\gamma_{i-1} < x < y < \gamma_i$. \qed

We may generalize the results of Theorem(3.1) as follows:

**Theorem 3.5.** Let there be given $r \geq 1$ sequences,

$$\Gamma^{(k)} = \{ \gamma_0^{(k)} = -\infty \leq \gamma_1^{(k)} \leq \cdots \leq \gamma_N^{(k)} < \infty = \gamma_N^{(k)} \}, \quad k = 1, \ldots, r,$$

that fulfill the conclusions of Theorem (3.1), that is,

$$\lambda_i \in (\gamma_{i-1}^{(k)}, \gamma_i^{(k)}), \quad k = 1, \ldots, r, \quad i = 1, \ldots, N.$$

Let us denote their union by $E$,

$$E = \{ e_0 = -\infty < e_1 \leq \cdots \leq e_{r(N-1)} < \infty = e_{r(N-1)+1} \}.$$

Then, either,

$$\lambda_i \in (e_{r(i-1)}, e_{r(i-1)+1}), \quad e_{r(i-1)} < e_{r(i-1)+1}$$

or $e_{r(i-1)} = \lambda_i = e_{r(i-1)+1}$.

We will prove this result using the following Lemma.
denotes their respective union. Then, either,

(9) \[ \lambda_i \in (\gamma_i-1, \gamma_i), \quad \gamma_i-1 < \gamma_i \]

or \( \gamma_i-1 = \lambda_i = \gamma_i \).

We will prove Theorem (3.1) using the following,

**Lemma 3.2.** Suppose there exist an index \( i \) such that,

(10) \[ \lambda = \gamma_i-1 = \gamma_i, \quad 2 \leq i \leq (N - 1). \]

Then \( \lambda \) is an eigenvalue of \( T \). We note that in this case, there exist indices \( r \) and \( s \) such that,

(11) \[ \theta_r = \gamma_i-1 = \gamma_i = \beta_s, \]

with \( 1 \leq r \leq k \), and \( 1 \leq s \leq m \), where \( m = N - (k + 1) \).

**Proof.** Let \( q_i(x) \) denote the characteristic polynomial of \( H_i \), then,

(12) \[ p(x) = -b^2_{k-1} p_{k-2}(x) q_{k+1}(x) + (x - a_k) p_{k-1}(x) q_{k+1}(x) - b^2_k p_{k-1}(x) q_{k+2}(x) \]
(13) \[ = q_{k+1}(x)[(x - a_k) p_{k-1}(x) - b^2_{k-1} p_{k-2}(x)] - b^2_k p_{k-1}(x) q_{k+2}(x) \]
(14) \[ = q_{k+1}(x) p_k(x) - b^2_k p_{k-1}(x) q_{k+2}(x), \]

where \( p_0(x) = q_N+1(x) \equiv 1 \), and \( q_N+2(x) \equiv 0 \). Hence,

(15) \[ p(\lambda) = q_{k+1}(\lambda) p_k(\theta_r) - b^2_k p_{k-1}(\lambda) q_{k+2}(\beta_s) = 0. \]

**Lemma 3.3.** There exist an index \( r \) such that,

(16) \[ \lambda = \theta_r, \quad 1 \leq r \leq k, \]

is an eigenvalue of \( T \), if and only if there exists an index \( s \), such that,

(17) \[ \lambda = \beta_s, \quad 1 \leq s \leq m. \]

**Proof.** Let, \( \lambda = \theta_r \) be an eigenvalue of \( T \), then,

(18) \[ 0 = p(\lambda) = q_{k+1}(\lambda) p_k(\theta_r) - b^2_k p_{k-1}(\theta_r) q_{k+2}(\lambda) = -b^2_k p_{k-1}(\theta_r) q_{k+2}(\lambda), \]

and \( b_k \neq 0 \), \( p_{k-1}(\theta_r) \neq 0 \) implies \( q_{k+2}(\lambda) = 0 \). The case \( \lambda = \beta_s \) is proved similarly.

**Corollary 3.4.** For the proof of Theorem (3.1) it is sufficient to show that there is at most one eigenvalue in each non-redundant interval.

**Proof.** We leave the proof for the reader.
matrix $T \in \mathcal{M}(N)$, by

$$
(2) \quad T = \begin{pmatrix}
    a_1 & b_1 & & & \\
    b_1 & a_2 & b_2 & & \\
    & \ddots & \ddots & \ddots & \\
    & & b_{N-1} & b_N & a_N
\end{pmatrix} = \begin{pmatrix}
    T_k & & & \\
    & b_k & & & \\
    & a_{k+1} & b_{k+1} & & \\
    & & b_{k+1} & & H_{k+2}
\end{pmatrix},
$$

where $T_k$ is the leading sub matrix of order $k$, and $H_{k+2}$ is the trailing sub matrix of order $N - (k + 1)$. We hereby assume that $T$ is unreduced, $b_i \neq 0$, $i = 1, \ldots, (N - 1)$. Hence, the eigenvalues of $T$, as well as those of its leading and trailing submatrices, are all simple, that is, of multiplicity one. We denote the extended set of eigenvalues of $T_k$ by

$$
(3) \quad \lambda^{(k)} = \{\lambda_0^{(k)} = -\infty < \lambda_1^{(k)} < \lambda_2^{(k)} < \cdots \lambda_{k-1}^{(k)} < \lambda_k^{(k)} < \lambda_{k+1}^{(k)} = \infty\},
$$

and its corresponding characteristic polynomial by

$$
(4) \quad p_k(x) = \text{det}(xI - T_k), \quad k = 1, \ldots, N,
$$

where we write for simplicity $\lambda_i$ and $p(x)$ for $k = N$. We now states and give a simple proof of the Cauchy interlacing theorem [11].

**Theorem 2.1.** The eigenvalues of $T_{k-1}$ interlace the eigenvalues of $T_k$

**Proof.** For $k = 1$ the proof is immediate, and in general,

$$
(5) \quad p_k(x) = (x - a_k)p_{k-1}(x) - b_{k-1}^2p_{k-2}(x).
$$

However, since by induction the eigenvalues of $T_{k-2}$ interlace those of $T_{k-1}$,

$$
(6) \quad \text{sign}(p_{k-2}(\lambda_i^{(k-1)} + )) \neq \text{sign}(p_{k-2}(\lambda_{i+1}^{(k-1)} - ))
$$

and $p_k(x)$ has a root inside each subinterval,

$$
(7) \quad (\lambda_i^{(k-1)}, \lambda_{i+1}^{(k-1)}), \quad 1 \leq i < k - 1.
$$

Finally, the same holds for the extreme subintervals as well. □

**3. Interlacing properties.** We present in this section a refinement of Cauchy theorem, which relates the eigenvalues of the matrix to the eigenvalues of its leading and trailing sub matrices.

**Theorem 3.1.** Let $T \in \mathcal{M}(N)$ be an unreduced symmetric tridiagonal matrix as in (2). Let $\theta$ and $\beta$ denote the extended set of eigenvalues of $T_k$, and $H_{k+2}$ respectively as in (3). Finally, let,

$$
(8) \quad \Gamma = \{\gamma_0 = -\infty < \gamma_1 \leq \cdots \leq \gamma_{N-1} < \infty = \gamma_N\},
$$
A NEW DIVIDE AND CONQUER PARALLEL ALGORITHM FOR COMPUTING THE EIGENVALUES OF A SYMMETRIC TRIDIAGONAL MATRIX

ILAN BAR-ON

Abstract. We present new interlacing properties for the eigenvalues of an unreduced tridiagonal symmetric matrix in terms of its leading and trailing sub matrices based on simple algebra and the Sylvester Inertia theorem. We then present a new divide and conquer parallel algorithm for computing the eigenvalues of a symmetric tridiagonal matrix. The new algorithm is more simple and straightforward than Cuppen’s method[6], and does not require the computation of the corresponding eigenvectors. We can then compute the set of eigenvalues in an interval in \( O(kn) \) time, where \( k \) is the number of eigenvalues, and \( n \) is the order of the matrix.

Key Words. Symmetric, Tridiagonal, Eigenvalues, Eigenvectors, Parallel algorithms.

1. Introduction. The interlacing properties of tridiagonal symmetric matrices are of interest for a variety of algorithms such as the Lanczos tridiagonalization method, and divide and conquer methods for computing the eigenvalues of the matrix. In this paper we present refined interlacing properties for the eigenvalues of a tridiagonal matrix in terms of the eigenvalues of its leading and trailing sub matrices, improving related results of Hill and Parlett[9][1]. We then derive a new divide and conquer parallel algorithm for computing the eigenvalues of the matrix, which does not require the computation of the eigenvectors of the matrix. Thus we can compute the set of eigenvalues in a given interval in \( O(kn) \) time where \( k \) is the number of eigenvalues, and \( n \) is the order of the matrix. Cuppen[6] divide and conquer algorithm on the other hand, is based on rank one updates[8][5][13] which require the computation of the corresponding eigenvectors and thus it is not efficient when only part of the spectrum is needed[7].

This paper is organized as follows. In Section 2, we present some definitions and notations. In Section 3, we present a refinement to the Cauchy interlacing theorem for unreduced symmetric tridiagonal matrices. Finally, in section 4, we present the new algorithm.

2. Basic definitions and notations. We denote by \( \mathcal{R}^n \), the set of real vectors of order \( n \), and the standard basis for this induced vector space by

\[ e_i, \quad i = 1, \ldots, n, \]

where \( e_i \) is all zeros beside the \( i \)th coordinate which is one. When needed we emphasize that a vector is in \( \mathcal{R}^n \) by writing for example \( e_i^{(n)} \). We denote by \( \mathcal{M}(n) \) the set of real matrices of order \( n \), and by \( A^T \) the transpose of \( A \). We denote a tridiagonal symmetric

* Technion - Israel Institute of Technology, Department of Computer Science, Technion City, Haifa 32 000, Israel. baron@cs.technion.ac.il