21, p.p. 201-208, 1974.

[BNS] Bunch, J. R., Nielsen, C.P., Sorensen, D.C., Rank-one modification of the symmetric eigenproblem, *Numer. Math.*, **31**, p.p. 31-48, 1978.

[Ca] Canny, J., Some algebraic and geometric computations in PSPACE, *Proc. of the 20th ACM Symposium on Theory of Computing,* p.p. 460-467, 1988.

[Cu] Cuppen, J.J.M., A divide and conquer method for the symmetric tridiagonal eigenproblem, *Numer. Math.*,**36**, p.p. 177-195, 1981.

[GL] Golub, G.H., van Loan, C.F., *Matrix Computations*, John Hopkins Univ. Press, Baltimore, Maryland, 1989.

[Pan 87] Pan, V., Complexity of parallel matrix computations, *Theoretical Computer Science,* **54**, pp. 65-85, 1987.

[P87] Pan, V., Sequential and parallel complexity of approximate evaluation of polynomial zeros, *Comput. and Math. (with Applications),* **14,8**, p.p. 591-622,1987.

[RT] Reif, J.H., Tate, S.H., Optimal size integer division circuits, *Proc. 21-st ACM Symp. on Theory of Computing,* p.p. 264-270, 1989.

[R] Renegar, J., On the worst-case arithmetic complexity of approximating zeros of polynomials, *J. of Complexity*, **3,2**, p.p. 90-113, 1987.

On the other hand, the eigenvalues of T_n are not very sensitive to the perturbation of the entries.

Theorem A2. *Let $\tilde{T}_n$ be an rst-matrix having diagonal and subdiagonal entries $\tilde{a}_i$, $i = 1,\ldots,n$, and $\tilde{b}_j$, $j = 1,\ldots,n-1$, respectively, such that $|\tilde{a}_i - a_i|, |\tilde{b}_j - b_j| \leq 2^{-\nu}$, $i = 1,\ldots,n$, $j = 1,\ldots,n-1$. Then for any eigenvalue λ_i of T_n, there exists an eigenvalue $\tilde{\lambda}_i$ of $\tilde{T}_n$ such that $|\tilde{\lambda}_i - \lambda_i| \leq 3(2^{-\nu})$.*

Proof. Theorem A.2 follows from Bauer-Fike theorem (see [GL], page 342) applied with the Euclidean norm, since $||\tilde{T} - T||_2 \leq 3(2^{-\nu})$. ∎

¿From the above theorems we deduce that the rst-matrix $\tilde{T}_n$ obtained by setting $\tilde{a}_i = \lceil 2^{\nu} a_i \rceil, \tilde{b}_j = \lceil 2^{\nu} b_j \rceil$ has integers entries with absolute values at most $2^{\nu+m+1.5}$, and its eigenvalues, divided by $2^{-\nu}$, yield approximations $\tilde{\lambda}_i$ to the eigenvalues of T_n such that $|\tilde{\lambda}_i - \lambda_i| \leq 3(2^{-\nu})$. Thus to insure the latter bound, it suffices to compute the entries of T_n with the precision of $\lceil \nu + m + 1.5 \rceil$ bits.

References.

[AHU] Aho, A.V., Hopcroft, J.E., Ullman, J.D., *The design and analysis of computer algorithms,* Addison Wesley, Mass, 1976.

[BFKT] Ben-Or, M., Feig, E., Kozen, D., Tiwari, P., A fast parallel algorithm for determining all roots of a polynomial with real roots, *SIAM J. on Computing,* **17**, p.p. 1081-1092, 1988.

[BKR] Ben-Or, M., Kozen, D., Reif, J., The complexity of elementary algebra and geometry, *Proc. of the 16th ACM Symposium on Theory of Computing,* p.p. 457-464, 1984 and *J.CSS,* **32**, p.p. 251-264, 1986.

[BOT] Ben-Or, M., Tiwari, P., Simple algorithm for approximating all roots of a polynomial with real roots, Research Report RC 14581, *IBM T.J. Watson Research Center*, Yorktown Heights, N.Y., 1989.

[BG] Bini, D., Gemigniani, L., On the Euclidean scheme for polynomials having interlaced real zeros, manuscript, 1990.

[BM] Borodin, A., Munro, I., *The computational complexity of algebraic and numeric problems*, American Elsevier, New York, 1975.

[Br] Brent, R.P., The parallel evaluation of general arithmetic expressions, *J. ACM,*

Acknowledgment. The authors thank Prasoon Tiwari for kindly supplying a copy of [BOT].

Appendix. Reduction of approximating to polynomial zeros to approximating to matrix eigenvalues. Let $p(x)$ be a polynomial of degree n having integer coefficients in the range from -2^m to 2^m and having only real zeros. Let T_n denote an rst-matrix whose characteristic polynomial is proportional to $p(x)$. We may compute the entries of such a matrix by applying the extended Euclidean scheme to $p(x)$ and $p'(x)$. The computation can be performed for the cost $O_A(n\log^2 n, 1)$ ([AHU]) or $O_A(\log^3 n, n^2)$ ([BG]). Combining these bounds with the cost of algorithm 6.1 implies the estimates $O_A(n\log^2 n(\log^2 b+\log n, 1)$ and $O_A(\log^2 n(\log^2 b+\log n), (n/\log b)^2)$ for the sequential and parallel arithmetic complexity of approximating to the zeros of a polynomial having only real zeros (compare [BOT]), and similarly, the sequential and parallel Boolean complexity estimates for approximating to the zeros of $p(x)$ can be reduced to ones for the extended Euclidean computations and for approximating to the eigenvalues of T_n. The estimates for the sequential Boolean complexity and for the parallel Boolean time of the latter stage dominate the respective estimates for the overall complexity of approximating to the zeros of $p(x)$, whereas the opposite is true for the bit-serial processor bound.

We will conclude with two theorems that will enable us to bound the precision of the entries of T_n remaining within a prescribed tolerance to the errors of approximating the eigenvalues of T_n. As before, let a_i, $i = 1, \dots, n$, and b_j, $j = 1, \dots, n-1$, denote the diagonal and the subdiagonal entries of T_n, respectively. We do not assume any bounds on a_i and b_j now, but will deduce them.

Theorem A1. *The entries of T_n have absolute values at most $2^{m+1.5}$.*

Proof. Due to the Cauchy theorem, the zeros of $p(x)$ have absolute values at most $1 + 2^m < 2^{m+1}$. Due to theorem 2.1, with $r = 1$, the diagonal entries of T_n, $a_1, \dots, a_n$, have absolute values at most 2^{m+1}. Applying theorem 2.1 with $r = 2$, we deduce that the eigenvalues of all the 2×2 submatrices $\begin{pmatrix} a_i & b_i \\ b_i & a_{i+1} \end{pmatrix}$ have absolute values at most 2^{m+1}, that is, $|a_i a_{i+1} - b_i^2| \le 2^{2m+2}$; whence $|b_i| \le 2^{m+1.5}$. ∎

size of the integers $Q(y_i)$ is bounded as follows:

$$|Q(y_i)| \leq 2^{n(2m+2u+3\log n+2)} = 2^{nk/2},\ k = O(m+u+\log n) \tag{7.1}$$

Now we apply the known algorithm for multipoint polynomial evaluation ([AHU], [BM]), where we use integer arithmetic modulo $p = 2^{nk/2}+1$ in the following way:

Recursively compute the coefficients of the monic polynomials $S_i^{(j)}$ of degrees 2^j,

$$\begin{aligned} S_k^{(0)} &= y - y_k\ ,\quad k = 1,\ldots,n\ , \\ S_i^{(j)} &= S_{2i-1}^{(j-1)} S_{2i}^{(j-1)},\ i = 1,\ldots,\frac{n}{2^j},\ j = 1,\ldots,\log n - 1\ . \end{aligned}$$

Recursively compute the coefficients of the polynomials $r_k^{(j)}$ of degrees at most $n/2^j$,

$$\begin{aligned} r_1^{(0)} &= Q(y), \\ r_{2i-1}^{(j)} &= r_i^{(j-1)} \bmod S_{2i-1}^{(\log n - j)}\ , \\ r_{2i}^{(j)} &= r_i^{(j-1)} \bmod S_{2i}^{(\log n - j)}\ , \end{aligned}$$

$i = 1,\ldots,2^{j-1}$, $j = 1,\ldots,\log n$, such that $r_k^{(\log n)} = Q(y_k)$, $k = 1,\ldots,n$. Combining the relations (7.1) and theorem 7.1 and recalling that $b = m+\mu$, we deduce that such a multipoint polynomial evaluation can be performed for the overall cost

$$O_A(\log^2 n \log\log n, n/\log\log n)$$

and, consequently,

$$O_B(\log^2 n \log\log n \log(nb) \log\log(nb)), n^2 b/\log\log n).$$

Therefore, algorithm 6.1 can be carried out for the cost

$$O_B(\log^2 n \log(nb)(\log^2 b + \log n) \log\log n \log\log(nb), n^2 b/\log\log n),$$

that is, it uses

$$O(n^2 b \log^2 n \log(nb)(\log^2 b + \log n) \log\log(nb))$$

bit-operations.

[AHU], p. 266, [RT]). Moreover, in $\mathbf{Z}_p$, $2^{jk} \neq 1 \bmod p$, $j = 1, \ldots, n-1$, and $2^{nk} = 1 \bmod p$, that is, 2^k is a primitive n-th root of 1, and any integer power of 2, in particular n, has its inverse in $\mathbf{Z}_p$. Therefore, in $\mathbf{Z}_p$, we may compute the product of a pair of n-th degree polynomials by means of 3 FFTs for the arithmetic cost $O_A(\log n, n)$ and, therefore, for the Boolean cost $O_B(\log n \log(nk) \log\log(nk), n^2 k)$. Finally, computing $q(x)$ and $r(x)$, we observe that, since $v(x)$ is monic, no divisions are needed, and the computation can be performed in $\mathbf{Z}_p$ by using the polynomial division algorithm of [RT]. We arrive at the cost bounds $O_A(\log n \log\log n, n/\log\log n)$ and, therefore,

$$O_B(\log n \ \log(nk) \ \log\log n \ \log\log(nk), n^2 k/\log\log n) \ .$$

■

Now, recall that the i-th stage of algorithm 5.1 (for the evaluation of the coefficients of the characteristic polynomial of T_n) consists in evaluating $8(n/2^i)$ products of pairs of polynomials of degrees at most 2^{i-1} and having integer coefficients whose absolute values are less than $2^{i+m2^{i+1}}$. Such coefficients are well defined by their values modulo $p = 2^{nk/2} + 1$, already for $k = O(mn)$. Therefore, we may use integer arithmetic modulo p, apply theorem 7.1, and deduce that the cost of algorithm 5.1 is given by

$$O_B(\log^2 n \log(nm) \log\log(mn), n^2 m).$$

Now consider the evaluation of $p(x)$ and $p'(x)$ at a set of points $x_1, x_2, \ldots, x_n$. We first observe that $|x_i| \leq 2^{m+2}$, since x_i, for $i = 1, \ldots, n$, are approximations, within the absolute errors at most 2^{-u}, to the eigenvalues of rst-matrices having the 1-norms at most $3(2^m)$. Moreover, we may consider x_i a binary value, $x_i = y_i/2^{u+\log n}$, where y_i is an integer, $|y_i| \leq 2^{m+2+u+\log n}$, since we are looking for approximation within the absolute error bound $2^{-u-\log n}$. Therefore, the evaluation of $p(x_i)$ and $p'(x_i)$ can be kept within the set of integers in the following way:

Consider $Q(y) = \sum_{i=1}^{n} \alpha_i 2^{(u+\log n)(n-i)} y^i$ where $p(x) = \sum_{i=o}^{n} \alpha_i x^i$. The polynomial $Q(y)$ has integer coefficients, each represented with at most $n(m+u+2\log n)$ binary digits, and satisfies the following relation: $Q(y) = 2^{(u+\log n)n} p(x)$, $y = 2^{(u+\log n)} x$. Moreover, the

since we need the output of algorithm 4.1 with a precision lower than b digits until we arrive at the stage of approximating the eigenvalues of the original matix T_n (see the details in [BOT]).

Remark 6.1. An equivalent version of algorithm 6.1 can be obtained by replacing the matrices $S_{\frac{n}{2}}$ and $R_{\frac{n}{2}}$ at stage 3 by the matrices $T_{\frac{n}{2}}$ and $\hat{T}_{\frac{n}{2}-1}$ of part b of theorem 2.2.

7. Bit-complexity estimates. We will prove that algorithm 6.1 can be implemented for the Boolean cost of $O_B(\log(nb)\log^2 n(\log^2 b + \log n)\log\log n, n^2(b+\log n)\frac{\log\log(bn)}{\log\log n})$, where $|a_i|, |b_i| \leq 2^m,\ b = m + u$.

The most expensive computations in algorithm 6.1 are the evaluation of the coefficients of the characteristic polynomial $p(x)$ of T_n at stage 1, performed by means of algorithm 5.1, and the evaluation at stage 3 of the values of $p(\lambda)$ and $p'(\lambda)$ at a set of n points, performed by using the multipoint polynomial evaluation algorithm of [AHU], [BM].

We will use the following auxiliary result:

Theorem 7.1. *Let n be a power of 2, k be a positive integer, $p = 2^{nk/2} + 1$, $u(x)$ and $v(x)$ be two polynomials with coefficients in $\mathbf{Z}_p$ (the set of integers with arithmetic modulo p), deg $v(x) = n_2 \leq$ deg $u(x) = n_1 = O(n)$. Then the coefficients of the polynomial*

$$w(x) = u(x)v(x),$$

can be computed modulo p for the Boolean cost $O_B(\log n \log(nk)\log\log(nk), n^2k)$. Moreover, if the polynomial $v(x)$ is monic, the coefficients of the two polynomials $q(x)$ and $r(x)$ such that

$$u(x) = v(x)q(x) + r(x)\ ,$$

degree$(r(x)) < n_2$, can be computed modulo p for the cost

$$O_B(\log n \log\log n \log(nk)\log\log(nk), \frac{n^2k}{\log\log n}).$$

Proof. We just need to combine the known estimates from [AHU] and [RT]. The Boolean cost of each arithmetic operation in $\mathbf{Z}_p$ is $O_B(\log d \log\log d, d)$. $d = \lceil \log p \rceil$ (see

though the number of processors grows, the numerical stability of the computations greatly improves this way.

6. The main algorithm. By using the tools and steps described in the preceding sections, we will now devise the main algorithm for the approximation to the eigenvalues of an $n \times n$ rst-matrix T with integer entries.

Algorithm 6.1.

Input: Integers u, m, n, $a_1, \ldots, a_n$, and nonzero integers $b_1, \ldots, b_{n-1}$, such that n is a power of 2 (compare Remark 2.1), $|a_i|, |b_i| \leq 2^m$. This input defines an rst-matrix T_n and the tolerance 2^{-u} to the output errors.

Output: $\gamma_1, \ldots, \gamma_n \in \mathbf{R}$ such that $|\gamma_i - \lambda_i| < 2^{-u}$, where $\lambda_i,\ i = 1, \ldots, n$, are the eigenvalues of T_n.

Stage 1: Compute the coefficients of the characteristic polynomial of T_n by applying algorithm 5.1.

Stage 2: Apply algorithm 6.1 to the input set

$$u+1\ ,\ m,\ n/2,\ a_1, \ldots, a_{\frac{n}{2}-1}, a_{\frac{n}{2}} - b_{\frac{n}{2}},\ b_1, \ldots, b_{\frac{n}{2}-1}\ ,$$

which defines an rst-matrix $S_{\frac{n}{2}}$, and to the input set

$$u+1\ ,\ m,\ n/2,\ \ a_{\frac{n}{2}+1} - b_{\frac{n}{2}}, a_{\frac{n}{2}+2}, \ldots, a_n,\ b_{\frac{n}{2}}, \ldots, b_{n-1},$$

which defines an rst-matrix $R_{\frac{n}{2}}$, thus obtaining approximations $\delta_1 \leq \delta_2 \leq \ldots \leq \delta_n$ to the eigenvalues of $S_{\frac{n}{2}}$ and $R_{\frac{n}{2}}$ within the absolute error bound $\Delta = 2^{-u-1}$. (The matrices $R_{\frac{n}{2}}$ and $S_{\frac{n}{2}}$ have been defined in theorem 2.3.)

Stage 3: Recall that the set of all the eigenvalues of $S_{\frac{n}{2}}$ and $R_{\frac{n}{2}}$ interleaves the set $\{\lambda_i\}$ (see theorem 2.3), set $d_i^+ = \delta_i + \Delta,\ d_i^- = \delta_i - \Delta$ and apply algorithms 3.1 and 4.1, to obtain $x_1, \ldots x_n$ such that $|x_i - \lambda_i| < 2^{-u}$.

The overall cost of the algorithm is $O_A(\log^3 n \log\log n(\log^2 b + \log n), n/\log\log n),\ b = u+m$, but this bound can be decreased to $O_A(\log^2 n \log\log n(\log^2 b + \log n),\ n/\log\log n)$,

so that

$$\begin{pmatrix} p_{i+1}(\lambda) \\ p_i(\lambda) \end{pmatrix} = F_i \dots F_0 \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \tag{5.2}$$

$$F_j = \begin{pmatrix} a_{j+1} - \lambda & -b_j^2 \\ 1 & 0 \end{pmatrix}, \quad j = 0, \dots, i \ , \ b_0 = 0 \ . \tag{5.3}$$

Now we will compute the coefficients of $p(\lambda)$ allowing the input entries of T_n from any field $\mathbf{F}$.

Algorithm 5.1.

Input: a positive integer $n = 2^h$ (compare remark 2.1), and $a_1, \dots a_n, b_1, \dots b_{n-1}$ (the entries of T_n, being the elements of a field $\mathbf{F}$).

Output: $\alpha_0, \dots, \alpha_n \in \mathbf{F}$, such that $p(\lambda) = \det(\lambda I - T_n) = \sum_{i=0}^{n} \alpha_i \lambda^i$.

Computation: first set $H_j^{(0)} = F_j$, $j = 0, \dots, n-1$; then for $i = 1, 2, \dots, \log n = h$, compute $H_j^{(i)} = H_{2j+1}^{(i-1)} H_{2j}^{(i-1)}$, $j = 0, \dots, 2^{-i}n - 1$, output the coefficients of the polynomials $p(\lambda) = (\ 1 \quad 0\ ) H_1^{(h)} \begin{pmatrix} 1 \\ 0 \end{pmatrix}$.

The i-th level of the computation is essentially reduced to $8n/2^i$ multiplications of the entries of the matrices $H_j^{(i)}$, which are polynomials of degrees at most 2^i, and to 4 additions of the products. Over the complex or real fields, $\mathbf{F}$, this means the $O_A(\log n, n)$ cost bound at each level i ([AHU], [BM]), that is, the overall cost of algorithm 5.1 is $O_A(\log^2 n, n)$.

Given the coefficients of $p(\lambda)$, and therefore, of $p'(\lambda)$, we may compute the values of both $p(\lambda)$ and $p'(\lambda)$ at a fixed set of n points for the cost $O_A(\log^2 n \log\log n, n/\log\log n)$ (see [AHU], [BM], [RT]).

Now recall that we apply Newton's iteration to approximate to the n eigenvalues of T_n. Each Newton's iteration step essentially amounts to computing the ratio $p(x)/p'(x)$ at n points, which now means the cost $O_A(\log^2 n \log\log n,\ n/\log\log n)$, and this is also an estimate for the cost of Algorithm 3.1.

Remark 5.1. Given the matrix T_n and a scalar x, $p(\lambda)$ can be evaluated at x for the cost $O_A(\log n, n)$. It is sufficient to apply algorithm 5.1 replacing λ by $\lambda + x$ and computing modulo λ. The same computation modulo λ^2 outputs the linear polynomial $p(x) + p'(x)\lambda$. Thus, given an n-by-n rst-matrix T_n, we may compute $p(\lambda)$ and $p'(\lambda)$ at $O(n)$ points for the cost $O_A(\log n, n^2)$, thus avoiding the evaluation of the coefficients of $p(\lambda)$. Even

either $\gamma_i - c_0 < 2\Delta$ or $\gamma_i - c_0 \leq \beta$ or $p(\gamma_i) \leq 0$. In the first case output $(c_0 + \gamma_i)/2$ and stop; in the second case, set $d_0 = \gamma_i$, $x_0 = (c_0 + \gamma_i)/2$ and go to stage 4; otherwise set $\beta = \gamma_i - c$ and go to the next stage.

Stage 3. Set $c_1 = \gamma_i$ and $d_1 = \gamma_{i-1}$. (Note that $d_1 - c_1 \geq c_1 - c_0 \geq 2\Delta$.) Compute $p((c_1 + d_1)/2)$. If $p((c_1 + d_1)/2) < 0$, then set $c_0 = (c_1 + d_1)/2$, $d_0 = d_1$ and go to the next stage (in this case, $\lambda \in [c_0, d_0]$ and $d_0 - c_0 < \min\{c_0 - c, d - d_0\}$). Otherwise set $c_0 = c_1$, $d_0 = (c_1 + d_1)/2$ and go to stage 2.

Stage 4. Apply $\lceil \log(5n^2) \rceil$ bisection steps to $[c_0, d_0]$ in order to find a starting point $x^{(0)}$ satisfying (4.1).

Stage 5. Apply $\lceil \log\log(0.8\frac{d_0 - c_0}{\Delta n^2}) \rceil$ Newton's steps (starting with $x^{(0)}$) in order to arrive at an approximation $\tilde{\lambda}$ to λ such that $|\tilde{\lambda} - \lambda| < \Delta$.

Since β only grows, whereas $d_0 - c_0$ decrases at least by two times, in each recursive call to stages 2 and 3, there can only be $O(\log^2(m + \log(\Delta^{-1})))$ such calls, and there can only be $O(\log(m + \log(\Delta^{-1})))$ Newton's iteration steps at stage 5 provided that $-3(2^m) < c < d < 3(2^m)$.

5. Computing the values of the characteristic polynomial and its derivative at a set of points. We will concurently apply algorithm 4.1 to all the selected intervals, each containing a single eigenvalue of T_n. At any of the $O(\log^2(m + \log(\Delta^{-1})))$ steps we will have to compute the values $p(\lambda)$ and $p'(\lambda)$ at a set of at most n points. In this section we will devise an algorithm for computing the values of the characteristic polynomial $p(\lambda) = p_n(\lambda) = \det(T_n - \lambda I)$ of T_n and of its first derivative at a given set of n points.

Due to the tridiagonal structure of the matrix T_n, we have the following three-term recurrence for the polynomials $p_i(\lambda) = \det(T_i - \lambda I)$:

$$\begin{aligned} &p_0(\lambda) = 1,\ p_1(\lambda) = a_1 - \lambda, \\ &p_{i+1}(\lambda) = (a_{i+1} - \lambda)p_i(\lambda) - b_i^2 p_{i-1}(\lambda),\ i = n-1, n-2, \ldots, 1, \end{aligned} \tag{5.1}$$

or in the equivalent matrix form,

$$\begin{pmatrix} p_{i+1}(\lambda) \\ p_i(\lambda) \end{pmatrix} = \begin{pmatrix} a_{i+1} - \lambda & -b_i^2 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} p_i(\lambda) \\ p_{i-1}(\lambda) \end{pmatrix},$$

4. Approximation to the eigenvalues by using Newton's iterations and the bisection method. In this section we will complement algorithm 3.1 by an algorithm that approximates (within a fixed absolute output error Δ) to an eigenvalue λ_j of T_n given a pair of real c and d, such that the interval $\mathcal{K} = \{x : \ c \le x \le d\}$ contains only this eigenvalue of T_n. We will apply the techniques of [BOT] and [R] based on the following result of [R]:

Theorem 4.1. *Let $p(x) = a_n \prod_{i=1}^{n}(x-\xi_i)$ be a polynomial. Let $x^{(0)} \in \mathbf{C}$ be such that $|x^{(0)}-\xi_1| < |x^{(0)}-\xi_2| \le \ldots \le |x^{(0)}-\xi_n|$. If $|x^{(0)}-\xi_1| < \frac{1}{5n^2}|x^{(0)}-\xi_2|$, then the sequence*

$$x^{(i+1)} = x^{(i)} - p(x^{(i)})/p'(x^{(i)}), \quad i = 0, 1, \ldots \ ,$$

generated by Newton's method, converges quadratically to ξ_1; moreover, $|x^{(i)}-\xi_1| \le 2^{3-2^i}|x^{(0)}-\xi_1|$.

If we have an initial approximation $c < x^{(0)} < d$ such that

$$|x^{(0)} - \lambda_j| \le \frac{1}{5n^2}\min\{x^{(0)}-c, d-x^{(0)}\}, \tag{4.1}$$

we may invoke theorem 4.1 and use Newton's iteration, thus arriving at the desired approximation to λ_j in $\log\log(0.8\frac{d-c}{\Delta n^2})$ Newton's steps. This observation leads us to the next procedure, which outputs a real point $\tilde{\lambda}$ such that

$$|\tilde{\lambda} - \lambda| \le \Delta, \quad \lambda = \lambda_j. \tag{4.2}$$

Algorithm 4.1.

Input: natural numbers $c < d$ such that $c < \lambda < d$ and a positive rational Δ.

Output: a rational $\tilde{\lambda}$ such that $|\lambda - \tilde{\lambda}| < \Delta$.

Stage 1. If $d - c < 2\Delta$ then set $\tilde{\lambda} = (c+d)/2$ and stop. Otherwise, set $\beta = 0$, compute $p(c)$ and $p((c+d)/2)$, set $c_0 = c$, $d_0 = d$ and restrict further computations to the interval $[c, \frac{c+d}{2}]$ if $p(c)\,p((c+d)/2) < 0$ and to the interval $[\frac{c+d}{2}, d]$ otherwise. Suppose, to be certain, that $p(c) < 0$, $p((c+d)/2) > 0$, and perform the following stages:

Stage 2. Apply the *bisection of the exponents* procedure to the interval $[c_0, (c_0 + d_0)/2]$, that is, successively evaluate $p(\gamma_i)$ for $\gamma_i = c_0 + (d_0 - c_0)2^{-2^i}$, $i = 1, 2, \ldots$, until

b) For every j, $j = 1, \ldots, n$, of the two consecutive candidate values ν_j and η_j, separated by the single interval $\mathcal{K}_j$, exactly $1 - c(\mathcal{K}_j)$ values should be selected as an approximation to the eigenvalue λ_j of T_n. (Indeed, $d_{j-1}^- < \nu_j < d_{j-1}^+ < d_j^- < \eta_j < d_j^+$ and $d_{j-1}^- \le \lambda_j \le d_j^+$, so that either $d_{j-1}^+ \le \lambda_j \le d_j^-$, and then $c(\mathcal{K}_j) = 1$, or $d_{j-1}^- \le \lambda_j \le d_{j-1}^+$, and then $|\nu_j - \lambda_j| \le \Delta$, or $d_j^- \le \lambda_j \le d_j^+$, and then $|\eta_j - \lambda_j| \le \Delta$.) The rule b implies that neither of the values ν_j and η_j should be selected if $c(\mathcal{K}_j) = 1$; otherwise, we choose exactly one of them as an approximation to λ_j.

Recursive application of the rules a and b completely defines the selection of the approximations to λ_j among all the candidate values η_j and ν_j for all j provided that we are given the values $p(d_i^+)$ and $p(d_i^-)$ for all i and that we know whether the leftmost interval $\mathcal{J}_{j,h}$ contains λ_j. The latter inclusion can be easily checked by using (3.1), (3.2) and Propositions 3.1-3.3. We shall decrease the parallel time of this selection process by using the following equivalent procedure (the equivalence can be easily verified by inspection):

To select an approximation to λ_j by η_j or by ν_j, first partition the set S of all the intervals $\mathcal{J}_{i,h}$ satisfying (3.8) into the maximal subsets whose consecutive intervals are only interleaved with intervals $\mathcal{K}_i$ having indices 0 and with intervals $\mathcal{J}_{i,h}$ having indices $h+1$. Number the intervals in each maximal subset from left to right by $1, 2, 3 \ldots$ and for each subset, proceed as follows: Let $N(i,h)$ denote the number assigned to the interval $\mathcal{J}_{i,h}$ in this numeration. Let $\delta = 0$ if there is no intervals $\mathcal{K}_j$ having index 1 to the left from the subset and let $\delta = 1$ otherwise. Then determine the indices of the intervals of the subset as follows:

$$c(\mathcal{J}_{i,h}) = h + 1 + (-1)^{\delta + N(i,h)}. \tag{3.9}$$

Select both η_i and ν_{i+h+1} as approximations within Δ to λ_i and λ_{i+h+1} if $c(\mathcal{J}_{i,h}) = h+2$ and select none of them if $c(\mathcal{J}_{i,h}) = h$ (according to the rule a).

Finally, for every i such that $c(\mathcal{J}_{i,h}) = h+1$, apply the rule b to select one of the η_i and ν_{i+h+1} as an approximation within Δ to λ_i or to λ_{i+h+1}, respectively.

Stage 2 (define the indices of the intervals $\mathcal{K}_j$ of (3.5) and $\mathcal{J}_{i,h}$ of (3.7)): Compute $p(d_i^-)$ and $p(d_i^+)$ for all i. Recall the relations (3.5)-(3.7), propositions 3.1 and 3.3 and corollary 3.1 and, for all i and j, define that

$$c(\mathcal{K}_j) = \begin{cases} 0 & \text{if } p(d_{j+1}^-)p(d_j^+) > 0 \\ 1 & \text{otherwise,} \end{cases}$$

$$c(\mathcal{J}_{i,h}) = h+1 \text{ if } \begin{cases} & \text{either } h \text{ is even and } p(d_i^-)p(d_{i+h}^+) < 0 \\ & \text{or } h \text{ is odd and } p(d_i^-)p(d_{i+h}^+) > 0, \end{cases}$$

otherwise,

$$c(\mathcal{J}_{i,h}) - h = \begin{cases} & 0 \\ & \text{or} \\ & 2 \end{cases} \tag{3.8}$$

Output the set of indices j such that $c(\mathcal{K}_j) = 1$, in which case $\lambda_{j-1} \in \mathcal{K}_j$.

Stage 3: (choosing approximations among the candidate values). By the beginning of this stage, for every j, $j = 1, \ldots, n$, it has been determined whether

$$\lambda_j \in \mathcal{K}_{j+1} \quad \text{(stage 2)},$$

or

$$|\lambda_j - \tilde{\lambda}_j| \leq 2\Delta \quad (\text{for } \tilde{\lambda}_j \text{ defined in stage 1}),$$

or, otherwise, at least one of the next two bounds hold:

$$|\lambda_j - \eta_j| \leq \Delta,$$
$$|\lambda_j - \nu_j| \leq \Delta \ .$$

It remains to distinguish between the two latter cases and to choose an approximation to λ_j by one of the two candidates η_j and ν_j. This process relies on the following simple rules:

a) For every interval $\mathcal{J}_{i,h}$, of the two candidate values η_i and ν_{i+h+1}, exactly $c(\mathcal{J}_{i,h}) - h$ values should be selected as approximations to λ_i and/or λ_{i+h} within Δ (due to the observations made at the end of stage 1). Since we know whether $c(\mathcal{K}_j) - h$ is odd or even, the rule a enables us to decide if we should select or not select one of the two candidate values η_i and ν_{i+h+1} as soon as we decided about another of them.

Proposition 3.2. *$d_i^- \le \lambda_{i+1} \le d_{i+1}^+$, for $i = 0, \ldots, n-1$.*

Corollary 3.1. *If $\mathcal{I}_i \cap \mathcal{I}_{i+1} \cap \ldots \cap \mathcal{I}_{i+h} \neq \emptyset$, then the points $\tilde{\lambda}_{i+j} = \frac{1}{2}(d_{i+j}^- + d_{i+j-1}^+)$ approximate to λ_{i+j} within the absolute error bound 2Δ for $j = 1, \ldots, h$. Moreover, $h \le c(\mathcal{I})$, where $\mathcal{I} = \{x : \ d_i^- \le x \le d_{i+h}^+\}$. Furthermore, $c(I) \le h+2$, if in addition, $I_{i-1} \cap I_i = I_{i+h} \cap I_{i+h+1} = \emptyset$.*

We will also use the following simple fact:

Proposition 3.3. *Let $p(a)p(b) \neq 0$, $\mathcal{J} = \{x : \ a \le x \le b\}$. Then $p(a)p(b) < 0$ if and only if $c(\mathcal{J})$ is odd.*

Algorithm 3.1.

Input: Positive rational Δ, an integer n and a set of rational numbers $D = \{d_i^-, d_i^+, \ i = 1, \ldots, n-1\} \cup \{d_0^+ = -3(2^{-m}), \ d_n^- = 3(2^m)\}$ such that (3.1), (3.2) and (3.6) hold, and $d_0^+ \le \lambda_1, \lambda_n \le d_n^-$ (compare (2.1)).

Output: For every j, $j = 1, \ldots, n$, either the interval $\mathcal{K}_j$ of (3.5) containing a unique eigenvalue λ_j of T_n or an approximation $\tilde{\lambda}_j$ to λ_i such that $|\lambda_j - \tilde{\lambda}_j| < 2\Delta$.

Stage 1 (form the union of the overlapping intervals): For $i = 0$ and for every i such that $d_{i-1}^+ < d_i^-$, determine the maximum $h = h(i)$ such that $d_{i+j-1}^+ \ge d_{i+j}^-$, for $j = 1, \ldots, h$, letting $h(i) = 0$ if $d_i^+ < d_{i+1}^-$. Output $\tilde{\lambda}_{i+j} = (d_{i+j}^- + d_{i+j-1}^+)/2$, $j = 1, \ldots, h$. (By virtue of corollary 3.1, $|\tilde{\lambda}_{i+j} - \lambda_{i+j}| \le 2\Delta$.) Save the values $\eta_i = (d_i^- + d_i^+)/2$, $\nu_{i+h+1} = (d_{i+h}^- + d_{i+h}^+)/2$ as candidates for approximating to λ_i and λ_{i+h+1}. Denote

$$\mathcal{J}_{i,h} = \{x : d_i^- \le x \le d_{i+h}^+\} = \bigcup_{j=0}^{h} \mathcal{I}_{i+j} \tag{3.7}$$

and observe that

$$\lambda_i \le d_i^+ = d_i^- + 2\Delta \ ,$$

$$\lambda_{i+h+1} \ge d_{i+h}^- \ge d_{i+h}^+ - 2\Delta \ ,$$

so that

$$|\lambda_i - \eta_i| \le \Delta \text{ if } \lambda_i \in \mathcal{J}_{i,h} \ ,$$

$$|\lambda_{i+h+1} - \nu_{i+h+1}| \le \Delta \text{ if } \lambda_{i+h} \in \mathcal{J}_{i,h} \ .$$

3. Computing the number of the eigenvalues in the intervals of nearly interlacing set. Let

$$d_0 < \lambda_1 \le d_1 \le \lambda_2 \le d_2 \le \ldots \le d_{n-1} \le \lambda_n < d_n \ . \tag{3.1}$$

Let pairs, $d_i^-,\ d_i^+$, of approximations to the real points d_i be given for $i = 1, \ldots, n-1$, such that for a fixed Δ,

$$d_i^+ - d_i^- = 2\Delta,\ d_i^- \le d_i \le d_i^+,\ i = 0, \ldots, n \ , \tag{3.2}$$

that is, the values d_i lie in the intervals

$$\mathcal{I}_i = \{x : \quad d_i^- \le x \le d_i^+\},\ i = 0, \ldots, n \ . \tag{3.3}$$

Let us be given $\Delta,\ d_i^-,\ d_i^+ = d_i^- + 2\Delta$ for $i = 0, \ldots, n$, and the black box subroutine for the exact evaluation of the value of the characteristic polynomial of T_n,

$$p(x) = \det(T_n - xI) \ . \tag{3.4}$$

In this section, for every eigenvalue λ_j of T_n (for $j = 1, \ldots, n$), either we will compute its approximation, within the absolute error bound 2Δ, or we will determine that the interval

$$\mathcal{K}_j = \{x : d_{j-1}^+ \le x \le d_j^-\}, \tag{3.5}$$

contains λ_j and no other eigenvalues of T_n. This problem was solved in [BOT]; we propose an alternate solution.

Without loss of generality, we assume that

$$p(d_i^-)p(d_i^+) \ne 0,\ i = 0, \ldots, n, \tag{3.6}$$

and we will use the next definition and simple auxiliary results implied by (3.1)-(3.6):

Definition 3.1. The number of the eigenvalues of T_n in a real interval $\mathcal{I}$ is called the *index* of $\mathcal{I}$ and is denoted $c(\mathcal{I})$.

Proposition 3.1. *Any interval $\mathcal{K}_j$ cannot contain more than one eigenvalue of T_n.*

Corollary 2.1 in a weaker form has been proven in [BFKT] (in terms of some zeros of the polynomials of the Euclidean scheme applied to the two polynomials $p(\lambda) = \det(\lambda I - T_n)$ and $p'(\lambda)$) and was used in order to solve the root-finding problem in NC for a polynomial having only real zeros. The result of part b has been proven in [BOT], still in terms of the zeros of the polynomials generated by the Euclidean scheme. The proofs in both papers [BFKT] and [BOT] are quite intricate, whereas the matrix formulation of these properties is a straightforward consequence of the Courant-Fischer minimax characterization, and the results can be immediately extended to polynomial zeros (as we pointed this out in the introduction).

Let us again apply the Courant-Fischer theorem in order to obtain still another set that interleaves the set Λ of the eigenvalues of T_n; this time we will rely on a suitable rank-one modification of the matrix T_n.

Theorem 2.3. *Let $\{\varphi_1 \le \varphi_2 \le \ldots \le \varphi_n\}$ be the set of all the eigenvalues of the matrices $S_k = T_k - \mathrm{Diag}(0,\ldots,0,b_k)$, $R_{n-k} = \hat{T}_{n-k} - \mathrm{Diag}(b_k,0,\ldots,0)$, where T_k and $\hat{T}_{n-k}$ are defined in theorem 2.2. Set $\varphi_{n+1} = \varphi_n + 2b_k$, $\varphi_0 = \varphi_1 + 2b_k$.*
If $b_k > 0$ then

$$\varphi_i \le \lambda_i \le \varphi_{i+1}, \quad i = 1,\ldots,n.$$

If $b_k < 0$ then

$$\varphi_{i-1} \le \lambda_i \le \varphi_i, \quad i = 1,\ldots,n.$$

Proof. Theorem 2.3 follows from theorem 8.1-5 of [GL], page 412, applied to the matrix equation

$$T_n = \mathrm{Diag}(T_k, \hat{T}_{n-k}) - 2b_k \mathbf{e}\mathbf{e}^T$$

where $\mathbf{e} = (e_i)$, $e_i = 1/\sqrt{2}$ for $i = k$, and $i = k+1$, $e_i = 0$ elsewhere. ∎

A different proof of theorem 2.3 is given in [Cu] and [BNS], where this theorem is used for separating the eigenvalues of an rst-matrix.

Theorem 2.2. *The eigenvalues of T_n satisfy the following relations:*

a) *If nk is a multiple of $2(k+1)$ and if $\{\mu_1 \le \mu_2 \le \ldots \le \mu_{\frac{k}{k+1}n}\}$ is the set of all the eigenvalues of the $k \times k$ principal submatrices $\tilde{T}_i$ of T_n, $i = 1, \ldots, \frac{n}{k+1}$, $\tilde{T}_i$ containing the (s,s) entries a_s of T_n for $s = (i-1)(k+1)+j$, $j = 1, \ldots, k$, then*

$$\lambda_{\frac{nk}{2(k+1)}} \le \mu_{\frac{nk}{2(k+1)}} \le \lambda_{\frac{n(k+2)}{2(k+1)}}.$$

b) *If $1 \le j \le n-2$ and if $\{\gamma_1 \le \gamma_2 \le \ldots \le \gamma_{n-1}\}$ is the set of all the eigenvalues of the two principal submatrices of T_n,*

$$T_j = \begin{pmatrix} a_1 & b_1 & & \\ b_1 & a_2 & \ddots & \\ & \ddots & \ddots & b_{j-1} \\ & & b_{j-1} & a_j \end{pmatrix}, \hat{T}_{n-j-1} = \begin{pmatrix} a_{j+2} & b_{j+2} & & \\ b_{j+2} & a_{j+3} & \ddots & \\ & \ddots & \ddots & b_{n-1} \\ & & b_{n-1} & a_n \end{pmatrix},$$

then

$$\lambda_i \le \gamma_i \le \lambda_{i+1},\ i = 1, \ldots, n-1.$$

In other words, part a) defines a splitting point of the level $(\frac{nk}{2(k+1)}, \frac{n(k+2)}{2(k+1)})$ for the set of the eigenvalues of T_n, and part b) defines an interlacing set for the set of the eigenvalues of T_n.

Proof. Theorem 2.2 immediately follows from theorem 2.1. In particular, to prove a), apply theorem 2.1 to the $r \times r$ principal submatrix of T_n (where $r = n - n/(k+1)$) obtained by deleting the $(i(k+1))$-th rows and columns of T_n for $i = 1, 2, \ldots, \frac{n}{k+1}$. This submatrix is the block diagonal matrix $\mathrm{Diag}(\tilde{T}_1, \ldots, \tilde{T}_{\frac{n}{k+1}})$. To prove part b), apply theorem 2.1 to the $(n-1) \times (n-1)$ principal submatrix of T_n obtained by deleting the $i-$th row and column of T_n; this submatrix is the 2×2 block diagonal matrix $\mathrm{Diag}(T_j, \hat{T}_{n-j-1})$. ∎

Remark 2.1. To extend part a) to the case of any n, apply theorem 2.2 to the tridiagonal matrix $\begin{pmatrix} T_n & 0 \\ 0 & \lambda I_s \end{pmatrix}$ for an appropriate $s < 2k+2$ and any λ.

Apply part a) of Theorem 2.2 for $k = 1$, then modify it by replacing $\tilde{T}_i$ by the 1-by-1 matrices (a_{2i}) for $i = 1, \ldots, \frac{n}{2}$, and thus arrive at

Corollary 2.1. *If n is a multiple of 4, $\{\sigma_1 \le \sigma_2 \le \ldots \le \sigma_{\frac{n}{2}}\} = \{a_{2i-1},\ i = 1, \ldots, \frac{n}{2}\}$, $\{\theta_1 \le \theta_2 \le \ldots \le \theta_{\frac{n}{2}}\} = \{a_{2i},\ i = 1, \ldots, \frac{n}{2}\}$, then*

$$\lambda_{\frac{n}{4}} \le \sigma_{\frac{n}{4}} \le \lambda_{\frac{3}{4}n}, \quad \lambda_{\frac{n}{4}} \le \theta_{\frac{n}{4}} \le \lambda_{\frac{3}{4}n}.$$

2. Interlacing sets and splitting points. Hereafter T_n denotes an $n \times n$ rst-matrix having diagonal entries $a_1, \ldots, a_n$, subdiagonal entries $b_1, \ldots, b_{n-1}$, and a set of n eigenvalues $\Lambda = \{\lambda_1 < \ldots < \lambda_n\}$. Without loss of generality, suppose that $b_1 \ldots b_{n-1} \neq 0$. Actually, if $b_j = 0$ for some j, the eigenvalue problem would be split into two eigenvalue problems of smaller dimensions. We assume that $|a_i|,\ |b_i| \leq 2^m$, m is a positive integer, so that, by virtue of Gerschgorin theorem,

$$-3(2^m) \leq \lambda_i \leq 3(2^m). \tag{2.1}$$

We say that the set $R = \{r_1, \ldots, r_k\}$ *strictly interleaves* the set $Q = \{q_1, \ldots, q_k\}$ and that R is an *interlacing set* for Q if $q_1 < r_1 < q_2 < \ldots < q_{k-1} < r_{k-1} < q_k < r_k$. Sometimes, we will let $r_k = +\infty$, that is, we will assume that $R = \{r_1, \ldots, r_{k-1}\}$. We say that s is a *splitting point of the level* (g, h) for the set Q if $q_g < s < q_h$.

In this section, we will prove some surprisingly simple corollaries from the Courant-Fischer minimax characterization ([GL], p. 411). Later on, they will lead us to simple algorithms for the evaluation of the interlacing sets and splitting points for the set of the eigenvalues of T_n. Using such sets and/or points, we may then devise divide-and-conquer algorithms for approximating to the eigenvalues of T_n, and we will specify such an algorithm in section 6.

Hereafter, $\mathrm{Diag}(B_1, \ldots, B_s)$ denotes the block diagonal matrix having diagonal blocks $B_1, \ldots, B_s$.

Theorem 2.1. *If A_r denotes an $r \times r$ principal submatrix of an $n \times n$ real symmetric matrix A, then the eigenvalues $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$ of A and the eigenvalues $\mu_1 \leq \mu_2 \leq \ldots \leq \mu_r$ of A_r satisfy the following relations:*

$$\lambda_i \leq \mu_i \leq \lambda_{i+n-r}, \quad i = 1, \ldots, r.$$

Proof. The latter relations hold for $r = n - 1$ (see corollary 8.1-4 to Courant-Fischer theorem in [GL], p. 411). In the general case, it is sufficient to apply these relations to a sequence $A = A_n, A_{n-1}, A_{n-2}, \ldots, A_r$ of principal submatrices of A such that A_i is a principal submatrix of A_{i+1}. ∎

We are ready to prove the following basis result:

this problem in [P87]). The problem is, of course, equivalent to computing in NC all the complex eigenvalues of an unsymmetric matrix.

To express the sequential and parallel cost of the algorithms, we will write $O_A(t,p)$ and $O_B(t,p)$ for the algorithms that require $O(t)$ parallel arithmetic steps using p processors and $O(t)$ parallel Boolean steps using p bit-serial processors, respectively. Note that $O_A(t,p)$ and $O_B(t,\ p)$ imply sequential arithmetic cost $O_A(tp,1)$ and sequential Boolean cost $O_B(tp,1)$, respectively, due to the slowdown principle of parallel computations (see [Br]). Under this notation, our algorithm supports approximating the eigenvalues of T_n for the arithmetic and Boolean parallel cost $O_A(\log^2 n(\log^2 b+\log n)\log\log n, n/\log\log n)$ and $O_B(\log^2 n\log(nb)(\log^2 b+\log n)\log\log n\log\log(nb), n^2 b/\log\log n)$, respectively, and for the sequential cost $O_A(n\log^2 n(\log^2 b+\log n),1)$ and $O_B(n^2 b\log^2 n\log(nb)(\log^2 b+\log n)\log\log(nb),1)$. For approximating all the n zeros of a polynomial $p(\lambda)$ having only real zeros, the same sequential cost bounds hold, but the parallel cost bounds change (in particular, to $O_A(\log^2 n(\log^2 b+\log n),(n/\log b)^2)$ under the arithmetic PRAM model) since we need to add the cost of the extended Euclidean algorithm supporting the transition from $p(\lambda)$ to T_n.

Below, all the logarithms are to the base 2.

We will organize the remainder of this paper as follows: In section 2 we deduce some properties of interlacing sets by using the Courant-Fischer theorem on the matrix eigenvalues. In section 3, for each eigenvalue of T_n, we compute either its approximation or an interval containing this eigenvalue but no other eigenvalues of T_n. In section 4, given an interval containing only one eigenvalue, we compute an approximation to this eigenvalue within the required precision by means of Newton's method and the bisection of the exponents. In section 5 we describe an algorithm for the simultaneous computation of the values of the characteristic polynomial and its derivative at a set of points. In section 6 we summarize the constructions of sections 2-5 and devise an algorithm for the approximation to all the eigenvalues. In section 7 we estimate the bit-complexity of the algorithm. Then, in the appendix we discuss the relations between the polynomial root finding problem and the problem of the computation of the eigenvalues of a matrix.

eigenvalues of T_n.

Technically, it may also be interesting that the Courant-Fischer minimax characterization may replace Sturm sequences in at least one more application. Namely, in [BFKT], the Sturm sequences have been used for computing a splitting point (rather than the interleaving set) for the set of the zeros of $p(\lambda)$; here too, the Courant-Fischer minimax characterization gives us a simple algorithm for computing such a splitting point for the eigenvalues of T_n, and this computation is more effective than one of [BFKT].

We also show (in section 2) another application of the Courant-Fischer minimax characterization: it may be used as the basis for the known class of divide-and-conquer algorithms for the eigenvalues of T_n (such algorithms have good potentials for practical implementation but not as good complexity estimates), and in this way, we *may extend our algorithm to computing the eigenvectors of* T_n (see [BNS] and [Cu]), although our approach does not require computing the eigenvectors if we only need to compute the eigenvalues.

With a set interleaving the zeros of $p(\lambda)$ (and the eigenvalues of T_n) available, the subsequent approximations to these zeros (and to these eigenvalues) are obtained by using more customary techniques of [BOT]; for the sake of completeness, we elaborate a modification of these techniques in sections 3 and 4 of our paper, which shows the correctness of the resulting algorithm.

Although our major concern here is about the computational complexity estimates, but on the other hand, if we cared more about the numerical stability and lower precision computations than about decreasing their asymptotic complexity, then we could have modified our main algorithm by directly computing the values of the characteristic polynomial, by means of our algorithm of Remark 5.1 and without computing the coefficients. In the resulting parallel NC algorithm, we only need by about $n/\log n$ times more processors, but its numerical stability is improved, which may make it competitive with the known alternate symmetric eigenvalue algorithms.

Our solution can be also interesting as a new basis for attacking a major open theoretical problem of computing all the complex zeros of a polynomial in NC (compare [BKR] and [Ca] and see the current record estimates for the parallel and sequential complexity of

may compute the coefficients of $p(\lambda)$ by applying even simpler an algorithm, which seems to be novel and which we present in section 5. We also show how this algorithm can be modified for computing the values of $p(\lambda)$ and $p'(\lambda)$ without computing their coefficients.

In the practice of computing, polynomials having only real zeros usually appear either as orthogonal polynomials or as the characteristic polynomials of Hermitian (or real symmetric) matrices. In both cases, there are simple reductions to rst-matrices (see [GL], [Pan 87]).

Our main result is a new parallel NC algorithm for approximating the eigenvalues of an rst-matrix T_n given its entries. We assume the customary arithmetic and Boolean PRAM models of parallel computation (where in each parallel step each processor performs one arithmetic or, respectively, Boolean operation), and we deduce near optimum upper bounds on parallel time and the number of processors required by our algorithm (by near optimum we mean upper bounds within polylogarithmic factors from the known lower bounds, which are straightforward lower bounds in our case). This is a dramatic improvement of the previous best parallel solution, based on the reduction of the problem to approximating the zeros of $p(\lambda)$ and on the solution of the latter problem by means of the algorithm of [BOT]. Our approach avoids using extended Euclidean computations invovled in the algorithm of [BOT], and this gives us the edge over [BOT].

The sequential version of our algorithm, as well as its extension to the sequential and parallel approximation to the zeros of $p(\lambda)$ given the coefficients, supports the same complexity estimates as the algorithm of [BOT] does, and in the sequential case, all these upper estimates are near optimum too.

Our approach, however, may be considered conceptually simpler and easier to comprehend. Specifically, the algorithm of [BOT] relies on computing the Sturm sequences and pseudo remainder sequences associated with $p(\lambda)$ and defining a set S of real points that interleaves the set of the zeros of $p(\lambda)$. Computing such a set S is the central (and also most innovative, most intricate and most involved) part of the algorithm of [BOT], and here we introduce our main innovation too (see section 2): we replace all the major techniques of [BOT] by some surprisingly simple corollaries from the Courant-Fischer minimax characterization theorem and immediately arrive at a set interleaving the set of the

COMPUTING MATRIX EIGENVALUES AND POLYNOMIAL ZEROS WHERE THE OUTPUT IS REAL.

*Dario Bini, Dipartimento di Matematica, Università di Pisa.**

*Victor Pan, Department of Computer Science, Columbia University, New York, NY 10027; Department of Mathematics and Computer Science, Lehman College, City University of New York and Department of Computer Science, State University of New York at Albany.***

CUCS-025-90

Abstract. Surprisingly simple corollaries from the Courant-Fischer minimax characterization theorem enable us to devise a very effective algorithm for the evaluation of a set S interleaving the set E of the eigenvalues of a real symmetric tridiagonal matrix T_n (as well as a point that splits E into two subsets of comparable cardinality). This is similar to the more complicated algorithms of [BOT] and [BFKT] that compute such interleaving sets and splitting points for the set of the zeros of a polynomial $p(\lambda)$ having only real zeros. As a result, the record upper estimates for the parallel and sequential complexity of approximating to all the zeros of such a polynomial are obtained in an alternate way and, moreover, are extended to the eigenvalue computation for T_n. Furthermore, we dramatically decreased the previous record upper estimates for the parallel complexity of the latter problem, to the level within polylogarithmic factors from the straightforward lower bounds.

1. Introduction. The study of polynomials having only real zeros is closely related to the study of real symmetric tridiagonal matrices (hereafter referred to as *rst-matrices*). Given the coefficients of an $n-th$ degree polynomial $p(\lambda)$ having only real zeros $\lambda_1, \ldots, \lambda_n$, we may compute the entries of an $n \times n$ rst-matrix T_n having the eigenvalues $\lambda_1, \ldots, \lambda_n$ by applying the extended Euclidean scheme to $p(\lambda)$ and $p'(\lambda)$, whereas for a given T_n, we

* Supported by NSF grant # 8805782 and by the Italian M.P.I. 40% funds.

** Supported by NSF grant # 8805782 and by PSC-CUNY Awards 668541 and 669210.