Electronic Transactions on Numerical Analysis.
Volume 17, pp. 195-205, 2004.

ISSN 1068-9613.

ETNA
Kent State University
etna@mcs.kent.edu

IMPROVED INITIALIZATION OF THE ACCELERATED AND ROBUST QR-LIKE POLYNOMIAL ROOT-FINDING*

DARIO A. BINI †, LUCA GEMIGNANI‡, AND VICTOR Y. PAN §

Abstract. We approximate polynomial roots numerically as the eigenvalues of a unitary diagonal plus rank-one matrix. We rely on our earlier adaptation of the QR algorithm, which exploits the semiseparable matrix structure to approximate the eigenvalues in a fast and robust way, but we substantially improve the performance of the resulting algorithm at the initial stage, as confirmed by our numerical tests.

Key words. QR iteration, eigenvalue computation, polynomial roots, semiseparable matrices, DFT, FFT, Moebius transformation.

AMS subject classifications. 65H17, 65F15.

1. Introduction. Polynomial root-finding is a fundamental mathematical problem with a long history [14]. The design of computationally effective polynomial root-finders is still an active research field. Besides the well-known applications to algebraic-geometric computations, we emphasize here the highly important applications in various areas of signal processing such as spectral factorization, filter and wavelet design, linear prediction, phase unwrapping, forming a cascade of lower order systems, etc. (see [10] and the references therein). In these contexts polynomials are typically generated by the z-transform of finite length signals, with polynomial order n equal to the number of sample points. Thus, orders of several hundred are common and, moreover, the coefficients are in general small with most of the roots located quite close to the unit circle.

Many root-finders are actually implemented as standard softwares in numerical libraries and environments such as NAG, IMSL, Mathematica[1] and Matlab[2]. Among the most used general purpose root-finders is the Jenkins-Traub method [9, 8] implemented by IMSL and Mathematica (`NSolve` function). It is instructive to compare it with another popular root-finder, which is employed by the function `roots` of Matlab and applies the matrix QR algorithm to compute the eigenvalues of the companion matrix associated with the given polynomial. Library implementations of the Jenkins-Traub method usually limit the acceptable degree of the input polynomial to 50. Numerical experiments reported in [10] confirm that the accuracy of the Jenkins-Traub program can dramatically deteriorate for relatively small degrees ($n = 50$) even if the roots of the input polynomial are numerically well-conditioned. On the contrary, the matrix approach based on the QR process yields a norm-wise backward stable root-finding algorithm [5, 18], which produces good results for most inputs. It has, however, a serious drawback: The resulting method is very space and time consuming ($O(n^2)$ and $O(n^3)$, respectively). Therefore, as Cleve Moler has pointed out in [11], this method may not be the best possible because "an algorithm designed specifically for polyno-

*Received February 25, 2004. Accepted for publication May 26, 2004. Recommended by A. Böttcher.
† Dipartimento di Matematica, Università di Pisa, Via Buonarroti 2, 56127 Pisa, Italy. E-mail: bini@dm.unipi.it. This work was partially supported by MIUR, grant number 2002014121, and by GNCS-INDAM.
‡Dipartimento di Matematica, Università di Pisa, Via F. Buonarroti 2, 56127 Pisa, Italy. E-mail: gemignan@dm.unipi.it. This work was partially supported by MIUR, grant number 2002014121, and by GNCS-INDAM.
§Department of Mathematics and Computer Science, Lehman College of CUNY, Bronx, NY 10468, USA. E-mail: vpan@lehman.cuny.edu. This work was supported by NSF Grant CCR 9732206 and PSC CUNY Awards 65393-0034.
[1]Mathematica is a registered trademark of Wolfram Research.
[2]Matlab is a registered trademark of The Mathworks, Inc..

ETNA
Kent State University
etna@mcs.kent.edu

mial roots might use order n storage and n^2 time."

In our present work we study numerically reliable algorithms which achieve this goal. For the sake of completeness, let us also mention the reduction in [4, 19] of polynomial root-finding to the tridiagonal eigenproblem and the algorithms in [12, 13] supporting the arithmetic and Boolean time complexity bounds for polynomial root-finding which are theoretically optimal up to polylogarithmic factors. We refer the reader to [14, 15, 17] for historical background and references. So far, however, such methods have been of no practical value because of their poor accuracy when implemented in finite precision arithmetic. Our present work is completely different. In this paper we propose a numerically reliable root-finding algorithm based on the exploitation of a different reduction to a matrix eigenvalue problem which is solved by the fast adaptation of QR iteration devised in [3]. To the advantage of our approach, we replace potentially unstable processes, such as the the construction of the tridiagonal matrix and the computation of its eigenvalues, by more robust computations while at the same time we keep the computational cost as low as possible.

A disturbing restriction of the algorithm in [3] is that it requires that the input matrix be a real diagonal plus rank-one matrix. The computation of a matrix in this form having a given characteristic polynomial leads to nontrivial numerical issues. In particular, the computed entries can be affected by large absolute errors thus resulting in poor accurate approximations of the eigenvalues. Moreover, the choice of real diagonal elements is also in conflict with the customary recipe of selecting the initial approximations on the complex circles centered at the origin [2]. This recipe is known to support faster convergence.

In this paper we propose and elaborate upon the following simple technique for circumventing the restriction in [3] on the input matrix. We assume that the input polynomial $p(z)$ is given by its degree n and by a black box for its evaluation at any point z. This setting is quite general since it covers many different polynomial representations without requiring to perform any possibly ill-conditioned basis conversion. We first evaluate $p(z)$ for $z = 0$ and at the n-th roots of unity (Fourier points) and then compute a unitary diagonal plus rank-one matrix $\widehat{A} = \widehat{D} + \widehat{\boldsymbol{u}}\widehat{\boldsymbol{v}}^H$, where $\widehat{D} = \mathrm{diag}[\xi_1, \ldots, \xi_n]$, $|\xi_i| = 1$ for all i, and $\widehat{A}$ has the characteristic polynomial $p(z)$. (We may generalize this approach by choosing $|\xi_i| = r$ for any fixed positive r.) Then we choose a Moebius (bilinear) transformation $\mathcal{M}(z) : \mathbb{C} \cup \{\infty\} \to \mathbb{C} \cup \{\infty\}$, $\mathcal{M}(z) = \dfrac{az - b}{cz - d}$, which maps the unit circle into the real axis and thus transforms the matrix $\widehat{A}$ into a real diagonal plus rank-one matrix $\mathcal{M}(\widehat{A}) = A = D + \boldsymbol{u}\boldsymbol{v}^H$, $D = \mathrm{diag}[\eta_1, \ldots, \eta_n]$ and $\eta_i \in \mathbb{R}$. We apply the algorithm in [3] to approximate the eigenvalues of the matrix A and then obtain the eigenvalues of $\widehat{A}$ (that is the roots of $p(z)$) simply by solving a linear equation in one variable for each eigenvalue. Apart from the application of the algorithm in [3], the computations essentially amount to evaluating $p(z)$ at the Fourier points. If we know the coefficients of $p(z)$ in the power form and if $n = 2^h$, then the evaluation cost is $O(n \log n)$ flops by using FFT's. The overall computational cost is dominated at the stage of the application of the algorithm in [3], which requires $O(n^2)$ flops for all roots (eigenvalues) (assuming a constant number of QR iterations per eigenvalue). Numerical experiments show that our root-finder exhibits a stable behavior.

We organize the paper as follows. In Sect. 2 we reduce the polynomial root-finding problem to solving an eigenvalue problem for a real diagonal plus rank-one matrix. In Sect. 3, for the sake of completeness, we describe our adaptation of the QR algorithm from [3]. In Sect. 4 we present and discuss the results of our extensive numerical experiments. Finally, conclusion and discussion are the subjects in Sect. 5.

ETNA
Kent State University
etna@mcs.kent.edu

2. An eigenvalue algorithm for computing polynomial roots. Assume that we seek numerical approximations of the roots of the n-degree polynomial $p(z)$,

$$p(z) = a_0 + a_1\, z + \ldots + a_n z^n = a_n \prod_{i=1}^{n} (z - \lambda_i), \quad a_n \neq 0, \quad a_i \in \mathbb{C}, \tag{2.1}$$

represented by means of a black box for computing the polynomial at any point z. Hereafter we write $\mathtt{i} = \sqrt{-1}$. By using the Lagrange interpolation formula applied on the nodes $\omega_j = \omega^j$, where $\omega = \exp(2\pi \mathtt{i}/n) \in \mathbb{C}$ is a primitive nth root of 1, we find that

$$p(z) - a_n z^n = (z^n - 1) \sum_{i=0}^{n-1} \frac{\omega_i (p(\omega_i) - a_n)}{n(z - \omega_i)}.$$

It follows that

$$p(z) = (z^n - 1)(a_n + \sum_{i=0}^{n-1} \frac{\omega_i p(\omega_i)}{n(z - \omega_i)}). \tag{2.2}$$

By evaluating both sides of this formula at the point $z = 0$, we obtain the following simple expression for the leading coefficients of $p(z)$,

$$a_n = \sum_{i=0}^{n-1} \frac{p(\omega_i)}{n} - p(0). \tag{2.3}$$

The root-finding problem for $p(z)$ given in the form (2.2) can be easily recasted into a matrix setting as the computation of the eigenvalues of a generalized companion matrix $\widehat{A} \in \mathbb{C}^{n \times n}$ associated with $p(z)$.

THEOREM 2.1. *Let $p(z)$ be the n-th degree polynomial (2.1) and denote by ω a primitive nth root of 1. Define the unitary diagonal plus rank-one matrix $\widehat{A}$ by*

$$\widehat{A} = \begin{bmatrix} 1 & & & \\ & \omega & & \\ & & \ddots & \\ & & & \omega^{n-1} \end{bmatrix} - \frac{1}{n a_n} \begin{bmatrix} p(1) \\ p(\omega) \\ \vdots \\ p(\omega^{n-1}) \end{bmatrix} \begin{bmatrix} 1 & \omega & \ldots & \omega^{n-1} \end{bmatrix}.$$

Then $a_n \det(zI - \widehat{A}) = p(z)$.

Proof. From the Sherman-Morrison-Woodbury formula (see [6], page 50) applied for the computation of the determinant of $zI - \widehat{A}$, we find that

$$a_n \det(zI - \widehat{A}) = a_n (z^n - 1)(1 + \frac{1}{a_n} \sum_{i=0}^{n-1} \frac{\omega_i p(\omega_i)}{n(z - \omega_i)}),$$

which coincides with $p(z)$ in the view of (2.2). □

An alternative derivation of this theorem relates the matrix $\widehat{A}$ to the Frobenius matrix F associated with $p(z)$,

$$F = \begin{bmatrix} 0 & & & & -a_0/a_n \\ 1 & 0 & & & -a_1/a_n \\ & 1 & 0 & & \vdots \\ & & \ddots & \ddots & \vdots \\ & & & 1 & -a_{n-1}/a_n \end{bmatrix}.$$

ETNA
Kent State University
etna@mcs.kent.edu

First observe that

$$F = \begin{bmatrix} 0 & & & 1 \\ 1 & 0 & & 0 \\ & \ddots & \ddots & \vdots \\ & & 1 & 0 \end{bmatrix} + \begin{bmatrix} -1 - a_0/a_n \\ -a_1/a_n \\ \vdots \\ -a_{n-1}/a_n \end{bmatrix} \begin{bmatrix} 0 & \dots & 0 & 1 \end{bmatrix} = Z + \boldsymbol{p}\boldsymbol{e}_n^T, \tag{2.4}$$

where $\boldsymbol{e}_n$ is the last column of the identity matrix I of order n. The *unit circulant* matrix Z can be diagonalized by the Fourier unitary matrix $\Omega = (\omega^{(i-1)(j-1)})$, where $V = \Omega/\sqrt{n}$, that is,

$$Z = V^H \operatorname{diag}[1, \omega, \dots, \omega^{n-1}]V = V^H \widehat{D} V.$$

By substituting this equation into (2.4), we obtain

$$F = V^H \widehat{D} V + \boldsymbol{p}\boldsymbol{e}_n^T = V^H(\widehat{D} + V\boldsymbol{p}\boldsymbol{e}_n^T V^H)V. \tag{2.5}$$

Since $\bar{\omega} = \omega^{-1}$, we deduce that

$$\boldsymbol{e}_n^T \Omega^H = \begin{bmatrix} 1 & \omega^{-(n-1)} & \dots & \omega^{-(n-1)(n-1)} \end{bmatrix} = \begin{bmatrix} 1 & \omega & \dots & \omega^{(n-1)} \end{bmatrix} = \widehat{\boldsymbol{v}}^T. \tag{2.6}$$

Furthermore, it is easily verified that

$$\Omega \boldsymbol{p} = a_n^{-1} \begin{bmatrix} p(1) & p(\omega) & \dots & p(\omega^{n-1}) \end{bmatrix}^T = \widehat{\boldsymbol{u}}. \tag{2.7}$$

Hence, Theorem 2.1 follows when we substitute (2.6) and (2.7) into the equation (2.5).

The advantage of the matrix formulation of the root-finding problem provided by Theorem 2.1 is that well established techniques of numerical linear algebra can be used to compute the eigenvalues of $\widehat{A}$. Because of its robustness, the QR iteration is usually the method of choice for finding the eigendecomposition of a matrix numerically [16, 6, 1].

REMARK 2.2. *Our second derivation of Theorem 2.1 shows that $\widehat{A}$ is similar by a unitary transformation to the Frobenius matrix F associated with $p(z)$. However, the conditioning of the root-finding problem for a polynomial $p(z)$ expressed by means of (2.2), (2.3) can be much different from that for the root-finding problem for the same polynomial given in the power form. We only point out that the sensitivity of the QR process applied to the matrices $\widehat{A}$ and F with respect to arbitrary (not structured) initial perturbations of the matrix entries should be comparable.*

Next we modify the matrix $\widehat{A}$ to speed up the computation of its eigenvalues by means of the QR algorithm. We introduce a matrix $A \in \mathbb{C}^{n\times n}$ related to $\widehat{A}$ and having the following features:

1. Each zero of $p(z)$ is obtained from a corresponding eigenvalue of A simply by solving a linear equation in one variable.
2. Approximations of the eigenvalues of A can be found by means of the QR algorithm in a fast and robust way.

Observe that the diagonal matrix $\widehat{D} = \operatorname{diag}[1, \omega, \dots, \omega^{n-1}]$ has entries located on the unit circle in the complex plane. A bilinear Moebius transformation $\mathcal{M}(z)$,

$$\mathcal{M}(z) \,:\, \mathbb{C} \cup \{\infty\} \to \mathbb{C} \cup \{\infty\},\ \mathcal{M}(z) = \frac{\delta z - \beta}{-\gamma z + \alpha}, \quad \alpha\delta - \beta\gamma \neq 0$$

for appropriate choices of the parameters α, β, δ and γ maps the unit circle into the real axis. If the matrix $\alpha I - \gamma \widehat{A}$ is nonsingular, then $A = \mathcal{M}(\widehat{A})$ is well defined and has the form

ETNA
Kent State University
etna@mcs.kent.edu

of a real diagonal plus rank-one matrix. We recall from [7] that the inverse of a Moebius transformation is still a Moebius transformation defined by

$$\mathcal{M}^{-1}(z) = \frac{\alpha z + \beta}{\gamma z + \delta}.$$

By combining this property with Theorem 4.3 of [20], we obtain the following simple result.

THEOREM 2.3. *Let $\gamma = |\gamma| \mathrm{e}^{\mathrm{i}\theta_\gamma}$ and $\delta = |\delta| \mathrm{e}^{\mathrm{i}\theta_\delta}$ be two arbitrary nonzero complex numbers such that $\mathrm{e}^{2\mathrm{i}(\theta_\gamma - \theta_\delta)} \neq 1$. Set $\alpha = |\gamma| \mathrm{e}^{\mathrm{i}\bar{\theta}}$, $\hat{\theta} = \tilde{\theta} + \theta_\gamma - \theta_\delta$, and $\beta = |\delta| \mathrm{e}^{\mathrm{i}\hat{\theta}}$. Then the function $\mathcal{M}(z) = \dfrac{\delta z - \beta}{-\gamma z + \alpha}$ is a Moebius transformation mapping the unit circle (except for the point $z = \alpha/\gamma$) onto the real axis.*

Assume now that $\mathcal{M}(z)$ is prescribed as in the previous theorem, $\alpha I - \gamma \widehat{A}$ is nonsingular and, moreover, $\mathcal{M}(\widehat{D})$ is well defined, i.e., $\omega^j \neq \alpha/\gamma$ for $j = 0, \ldots, n-1$. Then we have that

$$\mathcal{M}(\widehat{A}) = (\delta \widehat{A} - \beta I)(\alpha I - \gamma \widehat{A})^{-1},$$

which can be rewritten as

$$\mathcal{M}(\widehat{A}) = [(\delta \widehat{D} - \beta I) - \frac{\delta}{n a_n} \widehat{\boldsymbol{u}} \widehat{\boldsymbol{v}}^T][(\alpha I - \gamma \widehat{D}) + \frac{\gamma}{n a_n} \widehat{\boldsymbol{u}} \widehat{\boldsymbol{v}}^T]^{-1}, \tag{2.8}$$

where the diagonal matrix $\alpha I - \gamma \widehat{D}$ is invertible. From the Sherman-Morrison-Woodbury formula [6] it follows that

$$[(\alpha I - \gamma \widehat{D}) + \frac{\gamma}{n a_n} \widehat{\boldsymbol{u}} \widehat{\boldsymbol{v}}^T]^{-1} = (\alpha I - \gamma D)^{-1}(I - \theta \widehat{\boldsymbol{u}} \boldsymbol{v}^T),$$

where

$$\boldsymbol{v} = (\alpha I - \gamma \widehat{D})^{-1} \widehat{\boldsymbol{v}}, \quad \theta = \frac{\gamma}{n a_n + \gamma \boldsymbol{v}^T \widehat{\boldsymbol{u}}}. \tag{2.9}$$

Substitute the latter expression for the inverse of $(\alpha I - \gamma \widehat{D}) + \dfrac{\gamma}{n a_n} \widehat{\boldsymbol{u}} \widehat{\boldsymbol{v}}^T$ into (2.8) and deduce that

$$\mathcal{M}(\widehat{A}) = \mathcal{M}(\widehat{D}) - \theta \mathcal{M}(\widehat{D}) \widehat{\boldsymbol{u}} \boldsymbol{v}^T - \frac{\delta}{n a_n} \widehat{\boldsymbol{u}} \boldsymbol{v}^T + \frac{\delta \theta}{n a_n} \widehat{\boldsymbol{u}} (\boldsymbol{v}^T \widehat{\boldsymbol{u}}) \boldsymbol{v}^T,$$

which implies that

$$\mathcal{M}(\widehat{A}) = \mathcal{M}(\widehat{D}) - (\theta \mathcal{M}(\widehat{D}) \widehat{\boldsymbol{u}} + \frac{\delta}{n a_n} \widehat{\boldsymbol{u}} - \frac{\delta \theta}{n a_n} \widehat{\boldsymbol{u}} (\boldsymbol{v}^T \widehat{\boldsymbol{u}})) \boldsymbol{v}^T.$$

Write

$$\boldsymbol{u} = \theta \mathcal{M}(\widehat{D}) \widehat{\boldsymbol{u}} + \frac{\delta}{n a_n} \widehat{\boldsymbol{u}} - \frac{\delta \theta}{n a_n} \widehat{\boldsymbol{u}} (\boldsymbol{v}^T \widehat{\boldsymbol{u}}), \tag{2.10}$$

and finally arrive at the following simple representation of $\mathcal{M}(\widehat{A})$.

THEOREM 2.4. *Let $\widehat{A} = \widehat{D} - \dfrac{1}{n a_n} \widehat{\boldsymbol{u}} \widehat{\boldsymbol{v}}^T$ be the matrix of Theorem 2.1. Assume that $\mathcal{M}(z) = \dfrac{\delta z - \beta}{-\gamma z + \alpha}$ is a Moebius transformation determined as in Theorem 2.3 to map the*

ETNA
Kent State University
etna@mcs.kent.edu

unit circle onto the real axis in the complex plane. Moreover, suppose that $\alpha I - \gamma \widehat{A}$ *is nonsingular and* $\omega^j \neq \alpha/\gamma$ *for* $j = 0, \ldots, n-1$. *Then* $\mathcal{M}(A) = A = D - \boldsymbol{u}\boldsymbol{v}^T$ *is a real diagonal plus rank-one matrix, where*

$$D = \operatorname{diag}[\mathcal{M}(1), \mathcal{M}(\omega) \ldots, \mathcal{M}(\omega^{n-1})] \in \mathbb{R}^{n\times n},$$

and $\boldsymbol{u}$ *and* $\boldsymbol{v}$ *are defined by* (2.9) *and* (2.10)*, respectively.*

Each eigenvalue η_j of A is related to the corresponding eigenvalues λ_j of $\widehat{A}$ by $\eta_j = \mathcal{M}(\lambda_j) \neq -\delta/\gamma$. Once we have computed the eigenvalues of A we may retrieve those of $\widehat{A}$ simply by computing

$$\lambda_j = \frac{\alpha\eta_j + \beta}{\gamma\eta_j + \delta}, \quad 1 \leq j \leq n. \tag{2.11}$$

Based on the above results we devise an algorithm for approximating the roots of an nth degree polynomial $p(z)$ represented by means of a black box for its evaluation. The algorithm outputs the vector $\boldsymbol{\lambda}$ of the approximations to the roots of $p(z)$.

function [$\boldsymbol{\lambda}$] = **FastRoots**($\boldsymbol{p}$)
1) Evaluate $p(z)$ at the Fourier points $1, \omega, \ldots, \omega^{n-1}$, where $\omega = \cos(2\pi/n) + \mathtt{i}\sin(2\pi/n)$.
2) Compute the leading coefficient a_n of $p(z)$ by means of (2.3).
3) Form the vectors $\widehat{\boldsymbol{u}}$ and $\widehat{\boldsymbol{v}}$ defined in (2.6) and (2.7), respectively.
4) Choose random complex numbers γ and δ.
5) Choose a random real number $\tilde{\theta} \in [0, 1]$.
6) Define α and β as in Theorem 2.3.
7) Compute $(D)_{i,i} = \mathcal{M}(\omega^{i-1})$, for $i = 1, \ldots, n$.
8) Compute $\boldsymbol{u}$ and $\boldsymbol{v}$ by means of (2.9) and (2.10).
9) Compute approximations η_j of the eigenvalues of $D - \boldsymbol{u}\boldsymbol{v}^T$.
10) Approximate λ_j by using (2.11).

The core of this algorithm is the computation of the eigenvalues of the matrix $A = D - \boldsymbol{u}\boldsymbol{v}^T$. In the next section we recall our fast adaptation of the classical QR algorithm for finding the eigenvalues of a real diagonal plus rank-one matrix in a robust and fast way. Our root-finding method is obtained by incorporating such a variant of QR in the `FastRoots` procedure in order to carry out the eigenvalue computation at step 9 efficiently.

3. Fast QR iteration for real diagonal plus rank-one matrices. In this section for the sake of completeness we summarize the results of [3] for the computation of the eigenvalues of $n \times n$ generalized semiseparable matrices. Let us first specify this class of structured matrices, denoted by $\mathcal{C}_n$, by showing that it includes real diagonal plus rank-one matrices. A matrix $A = (a_{i,j}) \in \mathbb{C}^{n\times n}$ belongs to $\mathcal{C}_n$ if there exist real numbers $d_1, \ldots, d_n$, complex numbers $t_2, \ldots, t_{n-1}$, and four vectors $\boldsymbol{u} = [u_1, \ldots, u_n]^T \in \mathbb{C}^n$, $\boldsymbol{v} = [v_1, \ldots, v_n]^T \in \mathbb{C}^n$, $\boldsymbol{z} = [z_1, \ldots, z_n]^T \in \mathbb{C}^n$ and $\boldsymbol{w} = [w_1, \ldots, w_n]^T \in \mathbb{C}^n$ such that

$$\begin{cases} a_{i,i} = d_i + z_i\overline{w_i}, \quad 1 \leq i \leq n; \\ a_{i,j} = u_i t^{\times}_{i,j}\overline{v_j}, \quad 1 \leq j < i,\ 2 \leq i \leq n; \\ a_{i,j} = \overline{u_j} t^{\times}_{j,i} v_i + z_i\overline{w_j} - \overline{z_j} w_i, \quad 1 \leq i < j,\ 2 \leq j \leq n, \end{cases} \tag{3.1}$$

where $t^{\times}_{i,j} = t_{i-1} \ldots t_{j+1}$ for $i - 1 \geq j + 1$ and, otherwise, $t^{\times}_{i,i-1} = 1$. For $\boldsymbol{z} = \boldsymbol{u}$, $\boldsymbol{w} = \boldsymbol{v}$ and $t_i = 1$, $i = 2, \ldots, n-1$, then it is easily seen that $\mathcal{C}_n$ contains the real diagonal plus rank-one matrices of the form $A = D + \boldsymbol{u}\boldsymbol{v}^H$ with $D = \operatorname{diag}[d_1, \ldots, d_n] \in \mathbb{R}^{n\times n}$.

ETNA
Kent State University
etna@mcs.kent.edu

The computation of the eigenvalues of a generalized semiseparable matrix $A \in \mathcal{C}_n$ can be efficiently performed by means of the classical QR iteration with linear shift. The QR algorithm with linear shift applied to the matrix $A = A_0$ defines a sequence of similar matrices according to the following rule:

$$\text{(3.2)} \qquad \begin{cases} A_s - \sigma_s I_n = Q_s R_s, \\ A_{s+1} - \sigma_s I_n = R_s Q_s, \quad s \geq 0, \end{cases}$$

where Q_s is unitary, R_s is upper triangular, I_n denotes the identity matrix of order n and σ_s is a parameter called the *shift parameter*. The first equation of (3.2) yields a QR factorization of the matrix $A_s - \sigma_s I_n$. Under quite mild assumptions the matrix A_s tends to an upper triangular or, at least, a block upper triangular form thus yielding some desired information about the eigenvalues of A.

The following results are proved in [3]. The first of them states that the generalized semiseparable structure is invariant under the QR iterative process (3.2).

THEOREM 3.1. *Let A_s, $s = 1, \ldots, \tilde{s}+1$, be the matrices generated at the first $\tilde{s}+1$ iterations by the QR algorithm (3.2) starting with $A = A_0 \in \mathcal{C}_n$ of the form (3.1), where $R_0, \ldots, R_{\tilde{s}}$ are assumed to be nonsingular. Then, each A_s, with $0 \leq s \leq \tilde{s}+1$, belongs to $\mathcal{C}_n$. That is, for $0 \leq s \leq \tilde{s}+1$, there exist real numbers $d_1^{(s)}, \ldots, d_n^{(s)}$, $\boldsymbol{d}^{(s)} = [d_1^{(s)}, \ldots, d_n^{(s)}]^T \in \mathbb{R}^n$, complex numbers $t_2^{(s)}, \ldots, t_{n-1}^{(s)}$, $\boldsymbol{t}^{(s)} = [t_2^{(s)}, \ldots, t_{n-1}^{(s)}]^T \in \mathbb{C}^{n-2}$, and four n-vectors $\boldsymbol{u}^{(s)} = [u_1^{(s)}, \ldots, u_n^{(s)}]^T \in \mathbb{C}^n$, $\boldsymbol{v}^{(s)} = [v_1^{(s)}, \ldots, v_n^{(s)}]^T \in \mathbb{C}^n$, $\boldsymbol{z}^{(s)} = [z_1^{(s)}, \ldots, z_n^{(s)}]^T \in \mathbb{C}^n$ and $\boldsymbol{w}^{(s)} = [w_1^{(s)}, \ldots, w_n^{(s)}]^T \in \mathbb{C}^n$ such that $A_s = (a_{i,j}^{(s)})$ admits the following representation:*

$$\text{(3.3)} \qquad \begin{cases} a_{i,i}^{(s)} = d_i^{(s)} + z_i^{(s)} \overline{w_i^{(s)}}, \quad 1 \leq i \leq n; \\ a_{i,j}^{(s)} = u_i^{(s)} t_{i,j}^{(s)\times} \overline{v_j^{(s)}}, \quad 1 \leq j < i, \ 2 \leq i \leq n; \\ a_{i,j}^{(s)} = \overline{u_j^{(s)} t_{j,i}^{(s)}}^{\times} v_i^{(s)} + z_i^{(s)} \overline{w_j^{(s)}} - \overline{z_j^{(s)}} w_i^{(s)}, \quad 1 \leq i < j, \ 2 \leq j \leq n, \end{cases}$$

where $t_{i,j}^{(s)\times} = t_{i-1}^{(s)} \ldots t_{j+1}^{(s)}$ for $i-1 \geq j+1$ and, otherwise, $t_{i,i-1}^{(s)\times} = 1$.

The structured QR algorithm for generalized semiseparable matrices can be defined by a map Ψ,

$$\Psi : \mathbb{R}^n \times \mathbb{C}^n \times \mathbb{C}^n \times \mathbb{C}^n \times \mathbb{C}^n \times \mathbb{C}^{n-2} \times \mathbb{C} \to \mathbb{R}^n \times \mathbb{C}^n \times \mathbb{C}^n \times \mathbb{C}^n \times \mathbb{C}^n \times \mathbb{C}^{n-2}$$
$$(\boldsymbol{d}^{(s-1)}, \boldsymbol{u}^{(s-1)}, \boldsymbol{v}^{(s-1)}, \boldsymbol{z}^{(s-1)}, \boldsymbol{w}^{(s-1)}, \boldsymbol{t}^{(s-1)}, \sigma_{s-1}) \stackrel{\Psi}{\to} (\boldsymbol{d}^{(s)}, \boldsymbol{u}^{(s)}, \boldsymbol{v}^{(s)}, \boldsymbol{z}^{(s)}, \boldsymbol{w}^{(s)}, \boldsymbol{t}^{(s)}),$$

which, given a generalized semiseparable representation of A_{s-1} together with the value of the shift parameter σ_{s-1}, yields a generalized semiseparable representation of A_s satisfying (3.2). The next result is concerned with the complexity of such a map. Its proof in [3] is constructive and provides a fast implementation of the QR iteration (3.2) applied to the computation of the eigenvalues of a generalized semiseparable matrix $A = A_0$.

THEOREM 3.2. *Under the hypotheses of Theorem 3.1, there exists an algorithm which given an input $\boldsymbol{d}^{(s-1)}$, $\boldsymbol{u}^{(s-1)}$ $\boldsymbol{v}^{(s-1)}$, $\boldsymbol{z}^{(s-1)}$, $\boldsymbol{w}^{(s-1)}$, $\boldsymbol{t}^{(s-1)}$ and σ_{s-1} returns $\boldsymbol{d}^{(s)}$, $\boldsymbol{u}^{(s)}$ $\boldsymbol{v}^{(s)}$, $\boldsymbol{z}^{(s)}$, $\boldsymbol{w}^{(s)}$ and $\boldsymbol{t}^{(s)}$ as the output at the cost of $O(n)$ flops.*

If, for a fixed index $\hat{s}$, the matrices $R_{\hat{s}}$ and $A_{\hat{s}} - \sigma_{\hat{s}} I_n$ are singular, then $\sigma_{\hat{s}}$ is an eigenvalue of A_0, and a deflation technique should be employed. When we work in finite precision arithmetic, we also apply deflation if the entries of the matrix $A_{\hat{s}}$ satisfy a suitable stopping criterion. Let $A_{\hat{s}}[1 : n-k, 1 : n-k] \in \mathbb{C}^{(n-k)\times(n-k)}$ be the leading principal submatrix of $A_{\hat{s}}$ obtained from $A_{\hat{s}}$ by deleting its last k rows and columns. It is easily seen that $A_{\hat{s}}[1 : n-k, 1 : n-k]$ admits a representation similar to the one provided by Theorem 3.1. Such

ETNA
Kent State University
etna@mcs.kent.edu

a representation is found simply by truncating the corresponding representation of the matrix $A_{\hat{s}}$ of a larger size. Hence, $A_{\hat{s}}[1:n-k, 1:n-k] \in \mathcal{C}_{n-k}$ and, therefore, all the matrices generated by means of the QR scheme (3.2) applied to $A_0 \in \mathcal{C}_n$ for the computation of its eigenvalues still satisfy (3.3).

4. Numerical experiments. In this section we describe and discuss the results of our experiments in which we tested the speed and the accuracy of our root-finding algorithm. We have implemented in Matlab the function `FastRoots` described in the Section 2 and used it for computing the roots of polynomials of both small and large degree. At the step 9 of `FastRoots`, we applied the structured QR iteration for generalized semiseparable matrices devised in [3] and summarized in the previous section. To avoid possible unfortunate selections of the random parameters defining the Moebius transformation $\mathcal{M}(z)$, we repeated steps 4–8 twice and defined the final set of parameters that minimized the infinity norm of the corresponding matrix $D - \boldsymbol{u}\boldsymbol{v}^T$.

We tested the following polynomials, most of which are chosen or modified from [21] and [18]:

1. the "Wilkinson polynomial": $p(z) = \prod_{k=1}^{n}(z-k)$;
2. the scaled "Wilkinson polynomial": $p(z) = \prod_{k=1}^{n}(z-k/n)$;
3. the Chebyshev polynomial: $p(z) = \cos(n \arccos(z)),\ -1 \le \Re(z) \le 1$;
4. the monic polynomial with zeros equally spaced on the curve $z = x + \mathtt{i}\sin(\pi x)$, $-1 \le x \le 1$, namely $p(z) = \prod_{k=-n/2}^{n/2-1}(z - \frac{2(k+0.5)}{n-1} - \mathtt{i}\sin(\frac{2(k+0.5)}{n-1}))$;
5. the polynomial $p(z) = \prod_{k=1}^{2m-1}(z - \cos(\frac{\pi(k-m)}{2m}) - \mathtt{i}\sin(\frac{\pi(k-m)}{2m}))\prod_{k=2m}^{4m}(z - 0.9\cos(\frac{\pi(k-m)}{2m}) - 0.9\mathtt{i}\sin(\frac{\pi(k-m)}{2m}))$, $m = n/4$, which has a root distribution similar to the transfer function of an FIR low-pass filter [10];
6. the random polynomial $p(z) = (z^n - 1)(a_n + \sum_{i=0}^{n-1}\frac{\omega_i p(\omega_i)}{n(z-\omega_i)})$ with $p(\omega_i) =$ rand $+$ `i`rand, $p(0) =$ rand $+$ `i`rand.

Tables 4.1, 4.2, 4.3, 4.4 and 4.5 show the results of our numerical experiments for the polynomials from 1) to 5) by reporting the degree n, an estimate for the maximum condition number of the roots, the maximum, minimum and average absolute error of the computed root approximations over 100 experiments. Condition numbers are found from the Lagrange's representation of the polynomial $p(z)$ where we assume that the leading coefficient a_n is known exactly whereas the computed values $fl(p(\omega_i))$ satisfy $fl(p(\omega_i)) = p(\omega_i)(1+\epsilon_i)$, where $|\epsilon_i| \le n \cdot eps$ and eps is the machine precision. The poor results reported in Table 4.5 can be explained by observing the growth of the coefficients generated at intermediate steps by the Matlab function `poly` employed to compute the coefficients of the polynomial 5) given its zeros. Table 4.6 reports the output data for the polynomial 5) in the case where the values attained by the polynomial on the roots of unity are evaluated by using its factorization as a product of linear terms. Figure 4.1 covers our tests with random polynomials of the form 6) of high degree. It shows the $errors$ and the running $time$ for polynomials of degree $n(m) = 2^{2+m}$ for $m = 1, \ldots, 7$. Our test program returns these values as the output. The error value is computed as the maximum of the minimum distance between each computed eigenvalue and the set of "true" eigenvalues computed by the function `eig` of Matlab. For each size we carried out 100 numerical experiments. In each figure, the first plot reports the average value of the errors, and the second plot reports the ratio between the average values of running time for polynomials having degree $n(m)$ and $n(m+1)$. Since $m(n+1)/m(n) = 2$ and the proposed algorithm for computing all the zeros of $p(z)$ is expected to have a quadratic cost, this ratio should be close to 4 for large $n(m)$, which was indeed observed in these tests.

5. Conclusion. In this paper we have presented a novel root-finding algorithm based on eigenvalue computations which is appealing because of its memory requirements and com-

ETNA
Kent State University
etna@mcs.kent.edu

	Wilkinson polynomial			
n	cond	maxerr	minerr	average
10	1.61e+12	0.07	8.09e-05	0.004
20	6.64e+29	47.2	8.59	18.1

TABLE 4.1

	Scaled Wilkinson polynomial			
n	cond	maxerr	minerr	average
10	4.52e+07	2.42e-06	5.13e-10	5.32e-08
20	3.46e+16	0.4	0.05	0.12

TABLE 4.2

putational cost. By exploiting the structure of the associated eigenvalue problems enables us to yield a quadratic time using a linear memory space. The results of extensive numerical experiments confirm the robustness and the effectiveness of the proposed approach. The accuracy of computed results is generally in accordance with the estimates on the conditioning of polynomial roots for polynomials represented by means of an interpolation on the roots of unity.

REFERENCES

[1] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vost, *Templates for the solution of Algebraic Eigenvalue Problems: A Practical Guide*, SIAM, 2000.
[2] D. A. Bini and G. Fiorentino, *Design, analysis, and implementation of a multiprecision polynomial rootfinder*, Numer. Algorithms, 23 (2000), no. 2-3, pp. 127–173.
[3] D. A. Bini, L. Gemignani, and V. Y. Pan, *QR-like algorithms for generalized semiseparable matrices*, Technical Report 1470, Dipartimento di Matematica, Università di Pisa, Pisa, Italy, 2003.
[4] L. Brugnano and D. Trigiante, *Polynomial roots: the ultimate answer?*, Linear Algebra Appl., 225 (1995), pp. 207–219.
[5] A. Edelman and H. Murakami, *Polynomial roots from companion matrix eigenvalues*, Math. Comp., 64 (1995), no. 210, pp. 763–776.
[6] G. H. Golub and C. F. Van Loan, *Matrix Computations*, John Hopkins University Press, 1996.
[7] P. Henrici, *Applied and Computational Complex Analysis*, vol. 1, Wiley, 1974.
[8] M. A. Jenkins and J. F. Traub, *A three-stage variable-shift iteration for polynomial zeros and its relation to generalized Rayleigh iteration*, Numer. Math., 14 (1969/1970), pp. 252–263.
[9] M. A. Jenkins and J. F. Traub, *A three-stage algorithm for real polynomials using quadratic iteration*, SIAM J. Numer. Anal., 7 (1970), pp. 545–566.
[10] M. Lang and B. C. Frenzel, *Polynomial root-finding*, *IEEE Signal Process. Letters*, 1 (1994), no. 10, pp. 141–143.
[11] C. Moler, *Cleve's corner: Roots - of polynomials, that is*, The MathWorks Newsletter, 5 (1991), no. 1, pp. 8–9.
[12] C. A. Neff and J. H. Reif, *An efficient algorithm for the complex roots problem*. J. Complexity, 12 (1996), pp. 81–115.
[13] V. Y. Pan, *Optimal (up to polylog factors) sequential and parallel algorithms for approximating complex polynomial zeros*, in Proc. 27th Ann. ACM Symp. on Theory of Computing, ACM Press, New York, May 1995. pp. 741-750.
[14] V. Y. Pan, *Solving a polynomial equation: some history and recent progress*, SIAM Rev., 39 (1997), no. 2, pp. 187–220.
[15] V. Y. Pan, *Univariate polynomials: nearly optimal algorithms for numerical factorization and root-finding*, J. Symbolic Comput., 33 (2002), no. 5, pp. 701–733.
[16] B. N. Parlett, *The symmetric eigenvalue problem*, SIAM, 1998.
[17] S. Smale, *Complexity theory and numerical analysis* in Acta numer., vol. 6, Cambridge Univ. Press, Cambridge, 1997, pp. 523-551.

ETNA
Kent State University
etna@mcs.kent.edu

	Chebyshev polynomial			
n	cond	maxerr	minerr	average
10	1.34e+03	8.06e-12	3.09e-14	4.07e-13
20	6.6e+06	2.88e-06	1.67e-10	3.61e-08

TABLE 4.3

	$p(z) = \prod_{k=-n/2}^{n/2-1}(z - \frac{2(k+0.5)}{n-1} - \mathtt{i}\sin(\frac{2(k+0.5)}{n-1}))$			
n	cond	maxerr	minerr	average
10	54.8	1.03e-12	6.48e-15	4.05e-14
20	2.71e+04	2.90e-09	4.72e-11	9.58e-11

TABLE 4.4

	$p(z) = \prod_{k=1}^{2m-1}(z - \mathrm{e}^{\mathtt{i}\frac{\pi(k-m)}{2m}}) \prod_{k=2m}^{4m}(z - 0.9\mathrm{e}^{\mathtt{i}\frac{\pi(k-m)}{2m}}), m = n/4$			
n	cond	maxerr	minerr	average
20	7.18	1.64e-12	2.08e-13	3.25e-13
40	64.8	5.76e-09	5.16e-09	5.31e-09

TABLE 4.5

	$p(z) = \prod_{k=1}^{2m-1}(z - \mathrm{e}^{\mathtt{i}\frac{\pi(k-m)}{2m}}) \prod_{k=2m}^{4m}(z - 0.9\mathrm{e}^{\mathtt{i}\frac{\pi(k-m)}{2m}}), m = n/4$			
n	cond	maxerr	minerr	average
20	7.18	2.08e-12	4.26e-15	1.01e-13
40	64.8	5.16e-12	2.46e-14	2.2e-13

TABLE 4.6

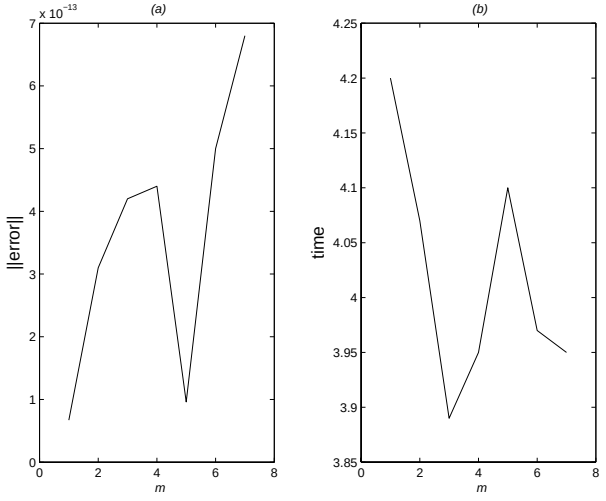

FIG. 4.1. *Random polynomials of degree* $n(m) = 2^{2+m}$, $1 \le m \le 7$.

ETNA
Kent State University
etna@mcs.kent.edu

[18] K. Toh and L. N. Trefethen, *Pseudozeros of polynomials and pseudospectra of companion matrices* Numer. Math., 68 (1994), no. 3, pp. 403–425.

[19] F. Uhlig, *General polynomial roots and their multiplicities in $O(n)$ memory and $O(n^2)$ time*, Linear Multilinear Algebra, 46 (1999), no. 4, pp. 327–359.

[20] M. Van Barel, D. Fasino, L. Gemignani, and N. Mastronardi, *Orthogonal rational functions*, Technical Report TW350, Katholieke Universiteit Leuven, Department Computerwetenschappen, 2002. Submitted to SIAM J. Matrix Anal. Appl..

[21] H. Zhang, *Numerical condition of polynomials in different forms*, Electron. Trans. Numer. Anal., 12 (2001), pp. 66–87.
http://etna.mcs.kent.edu/vol.12.2001/pp66-87.dir/pp66-87.pdf.