Structured Matrix Methods for Polynomial Root-Finding

Luca Gemignani
Department of Mathematics University of Pisa
Largo B. Pontecorvo 5, 56127 Pisa, Italy
gemignan@dm.unipi.it

ABSTRACT

In this paper we discuss the use of structured matrix methods for the numerical approximation of the zeros of a univariate polynomial. In particular, it is shown that root-finding algorithms based on floating-point eigenvalue computation can benefit from the structure of the matrix problem to reduce their complexity and memory requirements by an order of magnitude.

Categories and Subject Descriptors

G.1 [**Numerical Analysis**]: Numerical Linear Algebra; G.1.3 [**Numerical Linear Algebra**]: Eigenvalues and eigenvectors (direct and iterative methods)

General Terms

Algorithms

Keywords

Polynomial root-finding, eigenvalue computation, rank-structured matrices, complexity

1. INTRODUCTION

The problem of approximating the zeros of a univariate polynomial is classic in numerical analysis and computer algebra. Matrix methods based on eigenvalue computation are customary in both fields. They are proposed to the users of Matlab[1]: the internal function `roots` makes use of the QR algorithm for the computation of the eigenvalues of a companion matrix to approximate the zeros of a polynomial given by its coefficients. They are also at the core of the algorithm `eigensolve` in [18, 19] which iteratively determines estimates of the roots by solving a sequence of matrix eigenvalue problems. The matrix at the $k-$th step is a diagonal plus a rank-one matrix constructed starting from the approximations returned at the previous step. Numerical experiments reported in [19] say that this doubly iterative approach can outperform the best existing alternatives both in speed and accuracy. Specifically, experimental comparisons with the `mpsolve` algorithm developed by Bini and Fiorentino [4] show that `eigensolve` is faster in most situations, particularly on some ill-conditioned polynomials of high degree.

Companion and diagonal plus rank-one matrices belong to the class of *generalized companion* matrices [7]. A generalized companion matrix $A \in \mathbb{C}^{n\times n}$ of the polynomial $p(x)$ of degree n is such that $\det(A-xI) = 0$ precisely when $p(x) = 0$ and, moreover, the entries of A are determined from the coefficients of the representation of $p(x)$ w.r.t. some basis of the vector space of polynomials with complex coefficients and degree at most n. In the recent years fast adaptations of the QR algorithm have been designed (see [6, 1, 5, 2]) that are able to compute all the eigenvalues of certain $n \times n$ generalized companion matrices using linear memory space and linear time per iteration. The common requirement in these algorithms is that the input matrix can be represented as a small rank perturbation of an Hermitian or a unitary matrix in banded form. This condition is satisfied by at least three well-known classes of generalized companion matrices: real diagonal plus rank-one, symmetric tridiagonal plus rank-one and companion matrices. Nevertheless, it imposes a strong restriction on the choice of the polynomial basis used to represent the input polynomial.

The rationale for the approach in [18, 19] is that the iterative refinement of polynomial zeros can be complemented with the iterative improvement of the conditioning of the basis used to represent the input polynomial and this combination increases the accuracy of the output significantly. Ideally, if $p(x)$ has n pairwise distinct zeros then the best conditioned basis is some interpolatory (Newton or Lagrange) basis defined by the zeros of $p(x)$ which yields a generalized companion matrix A in diagonal or upper triangular form. Given n current approximations $\{s_1, \ldots, s_n\}$ of the zeros of $p(x)$ the algorithm `eigensolve` first computes the eigenvalues of the diagonal plus rank-one matrix generated by the coefficients of the Lagrange interpolating polynomial of degree $n-1$ of $p(x)$ on the nodes $s_1, \ldots, s_n$; then uses these eigenvalues to form the next estimates to the roots of $p(x)$. Unfortunately, the fast variants of the QR eigenvalue algorithm can not be incorporated into this procedure unless all the zeros of $p(x)$ are assumed to lie on the real axis or on the unit circle in the complex plane.

[1]Matlab is a registered trademark of The Mathworks, Inc..

ISSAC'07, July 29–August 1, 2007, Waterloo, Ontario, Canada.

In this paper we review the recent progresses concerning the fast adaptation of the QR algorithm for polynomial root-finding and propose an extension of these techniques to deal with a larger set of generalized companion matrices including matrices associated with certain interpolatory bases. The price for this advance is to replace the stable QR algorithm with a different iterative method using not orthogonal transformations. Generalizations of the classical LR and QR algorithms falling into the wider class of factorization/reverse order multiply eigenvalue algorithms were first studied by Della-Dora [9] and Elsner[16]. Algorithms of this kind are the SR, HR, QD, DQR and several other methods. Most of them are known to suffer from severe numerical instabilities. A remarkable exception is the DQR algorithm developed by Uhlig [23] which employs D-orthogonal transformations, where $D = \mathrm{diag}[1, \pm 1]$. The algorithm is proved to be (conditionally) stable whenever the magnitude of the transformations is properly monitored and huge intermediate quantities are avoided. D-orthogonal methods have found many other applications in numerical linear algebra, see e.g. [21] for an up-to-date survey.

As our major contribution we show that in the generic case the DQR algorithm can be used to efficiently compute the eigenvalues of a generalized companion matrix $A \in \mathbb{C}^{n\times n}$ for a polynomial represented in the Lagrange basis. This matrix has the form of a block diagonal matrix plus a rank-one correction, where the diagonal blocks are 1×1 or 2×2 matrices. A careful exploitation of the properties of the DQR iteration says that all the matrices generated by the algorithm can virtually be represented by a linear number of parameters. This allows one to implement the DQR iteration for computing the eigenvalues of A with $O(n)$ arithmetic operations (ops) per iteration and with $O(n)$ memory storage.

The paper is organized as follows. In Section 2 we give the background and review the theory around the fast variants of the QR algorithm for generalized companion matrices. In Section 3 we introduce some classes of generalized companion matrices for polynomials expressed in the Lagrange basis. The efficient eigenvalue computation of these matrices by means of fast adaptations of the DQR algorithm is the subject of Section 4. Finally, conclusions and possible further developments are drawn in Section 5.

2. FAST STRUCTURED QR ITERATION FOR GENERALIZED COMPANION MATRICES

Because of its robustness, the shifted QR iteration

$$\begin{aligned} A_1 &= A \\ A_k - \sigma_k I_n &= Q_k R_k \\ A_{k+1} &:= R_k Q_k + \sigma_k I_n, \end{aligned} \tag{1}$$

is usually the method of choice for finding the eigendecomposition of a matrix $A = A_1 \in \mathbb{C}^{n\times n}$ numerically. Here, Q_k is unitary and R_k is upper triangular for $k \geq 1$. The QR algorithm belongs to the class of factorization/reverse order multiply eigenvalue algorithms. Each step consists of first computing a QR factorization of the current iterate and then multiplying the unitary and the upper triangular factors in the reverse order to form the new iterate. The customary approach employs the QR eigenvalue algorithm complemented with a preliminary reduction of A to a simpler form B which makes the basic iteration computationally feasible. The cost of the QR iteration is $O(n^3)$ in general, but it decreases to $O(n)$ and $O(n^2)$ if B is Hermitian tridiagonal or upper Hessenberg, respectively.

Computationally appealing adaptations of the QR iteration have been recently devised which include the fast QR algorithms for tridiagonal and Hessenberg matrices as special cases. The advance considers the wider class of *rank-structured* matrices. Informally a matrix A has a rank structure if its off-diagonal blocks have small rank. A fundamental step in the design of fast algorithms to manipulate such kind of matrices is the description of the rank structure in terms of a small number of parameters that are called *generators*. The efficiency of linear algebra computations for rank-structured matrices can significantly be improved by working directly with the generators instead of the matrix entries.

To be more precise, let us assume that $A = (a_{i,j}) \in \mathbb{C}^{n\times n}$ satisfies the rank constrains

$$\begin{aligned} \max_{1\le k\le n-1} \mathrm{rank} A(k+1\colon n, 1\colon k) \le p, \\ \max_{1\le k\le n-1} \mathrm{rank} A(1\colon k, k+1\colon n) \le q, \end{aligned} \tag{2}$$

where $B(i\colon j, k\colon l)$ is the submatrix of B with entries having row and column indexes in the ranges i through j and k through l, respectively. Let $\mathcal{F}_{p,q}$ be the class of such $n\times n$ rank-structured matrices. Then $A \in \mathcal{F}_{p,q}$ can be represented by means of an asymptotically minimal set of generators as follows [11, 13, 14]:

$$a_{i,j} = \begin{cases} \boldsymbol{x}_i^T F_{i,j}^{\times} \boldsymbol{y}_j, & 1 \le j < i \le n, \\ \boldsymbol{z}_i^T G_{i,j}^{\times} \boldsymbol{w}_j, & 1 \le i < j \le n, \end{cases} \tag{3}$$

where $\boldsymbol{x}_{i+1}$, $\boldsymbol{y}_i$ and $\boldsymbol{z}_i$, $\boldsymbol{w}_{i+1}$, $1 \le i \le n-1$, are vectors of size p and q, respectively, and, moreover, $F_{i,j}^{\times} \in \mathbb{C}^{p\times p}$ and $G_{i,j}^{\times} \in \mathbb{C}^{q\times q}$ are defined by

$$F_{i,j}^{\times} = F_{i-1}\cdots F_{j+1},\ i > j+1; \quad F_{j+1,j} = I_p,$$

$$G_{i,j}^{\times} = G_{i+1}\cdots G_{j-1},\ j > i+1; \quad G_{i,i+1} = I_q,$$

for suitable matrices $F_2, \dots, F_{n-1} \in \mathbb{C}^{p\times p}$ and $G_2, \dots G_{n-1} \in \mathbb{C}^{q\times q}$. A matrix A given in the form (3) is called an order-(p,q)-*quasiseparable* matrix [13].

The quasiseparable structure provides a generalization of the *band* structure. If $F_i = F$, $G_i = G$ with $F^p = 0$, $G^q = 0$, then the matrix A defined by (3) is a band matrix with upper and lower bandwidth q and p, respectively. As for band matrices, each matrix $A \in \mathcal{F}_{p,q}$ can be stored in an array A_rkstru of size $O(n)$. The quasiseparable structure is partially maintained under arithmetic operations, inversion, LU and QR factorization since the matrices generated by these operations are still quasiseparable, but with a possibly different order of quasiseparability. This means that the class $\mathcal{F}_{p,q}$ is not closed under the QR iteration and, for a general quasiseparable matrix A its rank-structured property is rapidly lost after a few QR iterations. The invariance of the structure can however be easily proved under the further assumption that the input matrix A is a small rank perturbation of an Hermitian or a unitary matrix.

THEOREM 2.1. *Let us assume that $A \in \mathcal{F}_{p,q}$ is such that $A = B + U \cdot V^T$, where $U, V \in \mathbb{C}^{n\times r}$ and $B \in \mathcal{C}^{n\times n}$ is Hermitian and/or unitary. Then it follows that $A_k \in \mathcal{F}_{p,q+2r}$ for any matrix A_k, $k \geq 1$, generated by the shifted QR algorithm (1) applied to $A = A_1$.*

Specific instances of the Hermitian case were first considered in [6, 5, 17, 24] whereas the result for a general Hermitian matrix B can be found in [15]. References [1, 2, 8] are concerned with variants of the unitary case. The proof for a general unitary matrix B readily follows from the properties of unitary rank-structured matrices [3, 10, 20]

The previous theorem motivates the search for fast structured versions of the QR iteration which, given in input a condensed representation of A_k stored in the array A_k_rkstru together with the value of the shift parameter σ_k, return as output the array A_{k+1}_rkstru specifying the matrix A_{k+1} defined by (1). The next results are concerned with the complexity of such versions in two cases of special relevance for polynomial root-finding, namely, the case where A is real diagonal plus a small rank correction and the case where A is a companion matrix. The proofs are constructive and turn out immediately into practical QR eigenvalue algorithms for these kinds of matrices.

THEOREM 2.2. *[6] Let $A = diag[s_1, \ldots, s_n] + UV^T \in \mathcal{F}_{1,1}$, where $U, V \in \mathbb{C}^{n\times r}$ and $s_i \in \mathbb{R}$, $1 \leq i \leq n$. Then, for any $k \geq 1$ we have $A_k \in \mathcal{F}_{1,3}$ and, moreover, there exists an algorithm which given an input A_k_rkstru and σ_k returns A_{k+1}_rkstru as the output at the cost of $120n + O(1)$ ops.*

EXAMPLE 2.3. *In order to show that the fast implementation remains as robust as the customary QR algorithm consider the arrowhead matrix*

$$A = \begin{bmatrix} 1 & -1 & \ldots & -1 \\ 1 & \ddots & & \\ \vdots & & \ddots & \\ 1 & & & 1 \end{bmatrix}, \quad A^T = 2I - A.$$

The following table in [6] reports the estimated condition numbers and the distances between the set of the computed eigenvalues and the set of exact eigenvalues returned by the Matlab function `eig`. *These data are shown for the sizes $n = 2^m$, $m = 5, \ldots, 8$. Theoretically the condition number should be 1 due to the normality of A_1 but, in practice, the normality is lost after a few iterations so that the estimate provided by Matlab can be far from 1.*

n	*32*	*64*	*128*	*256*
cond	*3.8*	*2.4*	*7.2*	*1.2e+04*
err	*2.9e-15*	*6.7e-15*	*5.6e-14*	*1.5e-14*

THEOREM 2.4. *[2] Let $A = A_1 = H + \boldsymbol{u}\boldsymbol{v}^T$, where A is upper Hessenberg, H is unitary and $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{C}^n$. Then, for any $k \geq 1$ we have $A_k \in \mathcal{F}_{1,3}$ and, moreover, there exists an algorithm which given an input A_k_rkstru and σ_k returns A_{k+1}_rkstru as the output at the cost of $180n + O(1)$ ops.*

A different adaptation of the implicit QR algorithm for companion matrices is described in [8].

EXAMPLE 2.5. *The fast QR algorithm for companion matrices is applied for approximating the roots of the polynomial $p(z) = z^n - 1$ of degree n. Due to instabilities encountered in the deflation step the J-T (Jenkins and Traub) method implemented in Mathematica*[2] *has serious problems already for $n = 50$. The following table in [2] reports the value of err and cond for $n = 2^m$, $m = 7, 8, 9$.*

n	*128*	*256*	*512*
cond	*2.2e-16*	*2.2e-16*	*2.2e-16*
err	*5.2e-15*	*9.1e-15*	*1.7e-14*

3. GENERALIZED COMPANION MATRICES IN THE LAGRANGE BASIS

In this section we consider the problem of computing or refining numerical approximations of the zeros of a monic polynomial $p(z)$ of degree n

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

represented by means of a black box which can be queried for determining the value of the polynomial at any input. By using the Lagrange basis the root-finding problem for $p(z)$ can easily be recasted into a matrix setting as a matrix eigenvalue problem. Given n pairwise distinct nodes $s_1, \ldots, s_n$ define $w(z) = \prod_{i=1}^{n}(z - s_i)$ and let

$$p(z) - w(z) = \sum_{i=1}^{n} \frac{p(s_i)}{w'(s_i)} \prod_{j\neq i} (z - s_i)$$

be the Lagrange representation of the polynomial $p(z) - w(z)$ of degree $n - 1$. We have [22, 18]

THEOREM 3.1. *The diagonal plus rank-one matrix $A \in \mathbb{C}^{n\times n}$ given by*

$$A = \begin{bmatrix} s_1 & & & \\ & s_2 & & \\ & & \ddots & \\ & & & s_n \end{bmatrix} - \begin{bmatrix} \frac{p(s_1)}{w'(s_1)} \\ \frac{p(s_2)}{w'(s_2)} \\ \vdots \\ \frac{p(s_n)}{w'(s_n)} \end{bmatrix} \begin{bmatrix} 1 & 1 & \ldots & 1 \end{bmatrix} \tag{4}$$

is such that

$$\det(zI_n - A) = p(z).$$

In the case where $p(z)$ is a real polynomial, that is, $a_i \in \mathbb{R}$, and the interpolation nodes are either real or come in complex conjugate pairs then a real generalized companion matrix can also be defined. Suppose that $\lambda = \Re(\lambda) + \mathbf{i}\Im(\lambda) \in \mathbb{C}\backslash\mathbb{R}$, $w(z) = (z - \lambda)(z - \bar{\lambda})$ and consider the 2×2 real matrix

$$A = \begin{bmatrix} \Re(\lambda) & \Im(\lambda) \\ -\Im(\lambda) & \Re(\lambda) \end{bmatrix} - \begin{bmatrix} p \\ q \end{bmatrix} \begin{bmatrix} 1 & 1 \end{bmatrix},$$

[2]Mathematica is a registered trademark of Wolfram Research

where $p, q \in \mathbb{R}$ are to be determined in such a way that A has a prescribed characteristic polynomial $p(z) = a_0 + a_1 z + z^2$, with $a_1, a_2 \in \mathbb{R}$. We have

$$\det(zI_2 - A) = w(z) + (z - \Re(\lambda) - \Im(\lambda))p + (z - \Re(\lambda) + \Im(\lambda))q$$

from which it follows

$$p = \Re(\frac{p(\lambda)}{w'(\lambda)}) + \Im(\frac{p(\lambda)}{w'(\lambda)}), \quad q = \Re(\frac{p(\lambda)}{w'(\lambda)}) - \Im(\frac{p(\lambda)}{w'(\lambda)}).$$

The result generalizes to a real polynomial of degree $n \geq 2$.

THEOREM 3.2. *Let $S = \{s_1, \ldots, s_n\}$ with s_i distinct complex numbers, and let $w(z) = \prod_{i=1}^{n}(z - s_i)$. Assume that S can be partitioned into two disjoint sets $S_{\mathbb{C}} = \{s_1, \bar{s}_1, \ldots, s_p, \bar{s}_p\} \subset \mathbb{C} \setminus \mathbb{R}$ and $S_{\mathbb{R}} = \{s_{2p+1}, \ldots, s_n\} \subset \mathbb{R}$. For a given real monic polynomial $p(z)$ of degree n define the* Lagrange coefficients *of $p(z)$ on S by*

$$l_{2i-1} = \Re(\frac{p(s_i)}{w'(s_i)}) + \Im(\frac{p(s_i)}{w'(s_i)}),$$
$$l_{2i} = \Re(\frac{p(s_i)}{w'(s_i)}) - \Im(\frac{p(s_i)}{w'(s_i)}),$$

for $1 \leq i \leq p$, *and*

$$l_i = \frac{p(s_i)}{w'(s_i)}, \quad 2p + 1 \leq i \leq n.$$

Then the matrix $A \in \mathbb{R}^{n \times n}$,

$$A = D_1 \oplus D_2 - \begin{bmatrix} l_1 \\ \vdots \\ l_n \end{bmatrix} \begin{bmatrix} 1 & \ldots & 1 \end{bmatrix}, \quad (5)$$

where

$$D_1 = diag[\begin{bmatrix} \Re(s_1) & \Im(s_1) \\ -\Im(s_1) & \Re(s_1) \end{bmatrix}, \ldots, \begin{bmatrix} \Re(s_p) & \Im(s_p) \\ -\Im(s_p) & \Re(s_p) \end{bmatrix}]$$

and

$$D_2 = diag[s_{2p+1}, \ldots, s_n],$$

is such that

$$\det(zI_n - A) = p(z).$$

Matrices of the form (4) and (5) are referred to as generalized companion matrices in the Lagrange basis. Theorems 3.1 and 3.2 can be used to reduce the root-finding problem for $p(z)$ to a matrix eigenvalue problem for a generalized companion matrix A in the Lagrange basis. It is worth noting that the fast adaptations of the QR algorithm discussed in the previous section can not be applied to the input matrix A unless we assume some specific distribution of the zeros of $p(z)$. Indeed, A can be represented as a (block) diagonal matrix $T \in \mathbb{C}^{n \times n}$ modified by a rank-one correction but, generally, T is not Hermitian or unitary. However, T always satisfies the following *D-property*:

> There exists a diagonal matrix $D = \text{diag}[1, \pm 1]$ such that $D \cdot T = S$ and $S = S^T \in \mathbb{C}^{n \times n}$ is a complex symmetric matrix.

The exploitation of this property is at the core of the DQR eigenvalue algorithm for real tridiagonal matrices proposed in [23]. In the next section we apply this algorithm for the efficient eigenvalue computation of generalized companion matrices expressed in the Lagrange basis.

4. FAST STRUCTURED DQR ITERATION

For a given input matrix $A = A_1 \in \mathbb{C}^{n \times n}$ fulfilling the *D-property* the QR iteration can be suitably modified to incorporate the matrix $D = D^{-1}$ into the factorization/reverse order multiply scheme. Assume that the shifted matrix $A_1 - \sigma_1 I_n$ is factorized as

$$A_1 - \sigma_1 I_n = DQ_1 R_1,$$

where R_1 is upper triangular and Q_1 is $D-$orthogonal, i.e., $Q_1^T D Q_1 = D$. Then the next iterate A_2 is defined by

$$\begin{aligned} A_2 &= R_1 D Q_1 + \sigma_1 I_n \\ &= Q_1^{-1} D^{-1} D Q_1 R_1 D Q_1 + \sigma_1 I_n \\ &= Q_1^{-1} D^{-1} A_1 D Q_1 \\ &= D^{-1} (Q_1^T D) D A_1 (D Q_1). \end{aligned}$$

Observe that A_2 is similar to A_1 and, moreover, it still satisfies the *D-property* for the same matrix D.

When applicable the shifted DQR algorithm [23] applied to $A = A_1$ proceeds by generating a sequence of matrices $\{A_k\}$ as follows:

$$\begin{aligned} & A_1 = A \\ & A_k - \sigma_k I_n = D Q_k R_k \\ & A_{k+1} := R_k D Q_k + \sigma_k I_n, \end{aligned} \quad (6)$$

where R_k and Q_k are upper triangular and $D-$orthogonal, respectively. Under quite mild assumptions A_k approaches a diagonal or a block diagonal form thus yielding approximations of the eigenvalues of A. A convergence proof can be found in [23].

The DQR algorithm can break down due to the use of D-orthogonal factorizations. The reduction of a matrix A into an upper triangular form using $D-$orthogonal transformations can be accomplished by a sequence of 2×2 elementary matrices in a process akin to the use of Givens rotations when employed in unitary elimination schemes. It is found that $\text{diag}[1, -1]-$orthogonal matrices have the form

$$\begin{bmatrix} a & b \\ b & a \end{bmatrix}, \quad a^2 - b^2 = 1, \; a, b \in \mathbb{C},$$

whereas I_2-orthogonal matrices can be specified as

$$\begin{bmatrix} a & b \\ -b & a \end{bmatrix}, \; a^2 + b^2 = 1, \; a, b \in \mathbb{C}.$$

The zeroing process by means of a $\text{diag}[d_1, d_2]-$orthogonal matrix applied to the vector $[x, y]^T$ can only break down if

$$x^2 = y^2 \;\; \text{for} \;\; d_1 d_2 = -1, \quad (7)$$

and

$$x^2 = -y^2 \;\; \text{for} \;\; d_1 d_2 = 1. \quad (8)$$

The occurrence of breakdowns in the DQR iteration can lead to potential accuracy problems in finite precision arithmetic.

We postpone the discussion of some numerical issues related to the implementation of the DQR technique to the next section.

Now let us analyze the properties of the DQR iteration (6) applied for computing the eigenvalues of an input matrix $A = A_1 = T + \boldsymbol{u}\boldsymbol{v}^T$, where $T \in \mathcal{F}_{1,1}$ meets the *D-property.* Our main result is the following.

THEOREM 4.1. *If the sequence of nonsingular DQR iterates $\{A_k\}$ is generated by* (6) *starting from $A = A_1 = T + \boldsymbol{u}\boldsymbol{v}^T$, where $T \in \mathcal{F}_{1,1}$ satisfies the* D-property *for a certain $D = diag[1, \pm 1]$, then $A_k \in \mathcal{F}_{2,4}$ for any $k \geq 1$.*

PROOF. By induction. Let us assume that $A_k \in \mathcal{F}_{2,4}$ and, moreover, $A_k = T_k + \boldsymbol{u}_k\boldsymbol{v}_k^T$ where $T_k \in \mathbb{C}^{n\times n}$ fulfills the *D-property* for the same matrix D as T_1. Since A_k is nonsingular we can write

$$A_{k+1} = R_k D Q_k R_k R_k^{-1} + \sigma_1 R_k R_k^{-1} = R_k A_k R_k^{-1}.$$

Because of the upper triangular form of R_k the similarity transformation maintains the rank constrains in the strictly lower triangular part of A_k and, therefore,

$$\max_{1\leq i\leq n-1} \operatorname{rank} A_{k+1}(i+1\colon n, 1\colon i) \leq 2.$$

The rank structure in the strictly upper triangular part of A_{k+1} follows from the relation

$$DA_{k+1} = (Q_k^T D) D (T_k + \boldsymbol{u}_k\boldsymbol{v}_k^T)(DQ_k)$$

which says that DA_{k+1} can be represented as the symmetric matrix $S_k = (Q_k^T D) D T_k (DQ_k)$ plus a rank-one correction. Specifically, we obtain that $S_k \in \mathcal{F}_{3,3}$ and, hence, DA_{k+1} and, a fortiori, A_{k+1} belong to $\mathcal{F}_{2,4}$. The proof is completed by observing that

$$A_{k+1} = DS_k + \boldsymbol{u}_{k+1}\boldsymbol{v}_{k+1}^T = T_{k+1} + \boldsymbol{u}_{k+1}\boldsymbol{v}_{k+1}^T.$$

$\square$

This theorem enables the extension to the DQR iteration of the computational results concerning the QR eigenvalue algorithm applied to an input rank-structured matrix. As an example we obtain the following generalization of Theorem 2.2.

THEOREM 4.2. *Let $A = diag[s_1, \ldots, s_n] + UV^T \in \mathcal{F}_{1,1}$, where $U, V \in \mathbb{C}^{n\times r}$. Assume that the sequence of nonsingular DQR iterates $\{A_k\}_{k\geq 1}$ generated by the DQR algorithm* (6) *applied to $A = A_1$ is well defined. Then for any $k \geq 1$ we have $A_k \in \mathcal{F}_{1,3}$ and, moreover, there exists an algorithm which given an input A_k_rkstru and σ_k returns A_{k+1}_rkstru as the output at the cost of $120n + O(1)$ ops.*

A similar generalization also holds in the case where the input matrix is a rank-one correction of a block diagonal matrix satisfying the *D-property.* Such generalizations can be applied for the solution of the matrix eigenvalue problems involving generalized companion matrices expressed in the Lagrange basis at the overall cost of $O(n^2)$ ops and $O(n)$ memory storage.

5. CONCLUSIONS AND FURTHER DEVELOPMENTS

Matrix eigenvalue problems arising in polynomial root-finding are typically structured. The search for structures which can be efficiently treated is a prominent issue. We have shown that it is possible to enlarge the set of eligible structures to include eigenvalue problems for generalized companion matrices in the Lagrange basis by replacing the customary QR iteration with the DQR iteration. The price for using not orthogonal transformations is the potential occurrence of breakdowns and numerical instabilities. The stable QR schemes should be used whenever unavoidable problems occurred. Extensive numerical experiments are still required to achieve statistical evidence of the practical behaviour of the fast DQR-based procedures.

An error analysis for the DQR algorithm applied for eigenvalue computation of real tridiagonal matrices was performed in [23]. The conclusion is that, when applicable, the DQR iteration is conditionally stable provided that a simple check on the magnitude of the D-orthogonal transformations is incorporated into the algorithm. In particular, the analysis shows that if we limit the size of the entries of $D-$orthogonal transformations to below 10^p then we could guarantee about $m - 4p$ accurate eigenvalue digits of the m being carried. In view of (7) and (8) the limit is only exceeded if some entries in the DQR step agree in the first nonzero k digits with $k \geq 2p - 1$. Thus for sufficiently large m the computation would just about never break down.

Our very preliminary numerical experience with generalized companion matrices confirm these claims and suggests the use of the fast DQR iteration in a multiprecision floating point environment. For the sake of example, let us consider the 3×3 arrowhead matrix

$$A = A_1 = \begin{bmatrix} 1 & -1 & -1 \\ 1.00005 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}.$$

The eigenvalues of A_{11} computed in Matlab after having performed 10 unshifted DQR steps with $D = \mathrm{diag}[1, 1, -1]$ match those of the input matrix A_1 in their first 8 significant digits. The maximal element of the hyperbolic transformations has size about 100.

$D-$orthogonal methods can also be used to transform a given rank-structured matrix $A \in \mathbb{C}^{n\times n}$ to upper Hessenberg form B, as a means of finding the eigenvalues of the matrix efficiently. Such reduction techniques are generalizations of the algorithms presented in [12] and can be applied to a larger class of input rank-structured matrices at the same computational cost of $O(n^2)$ arithmetic operations. Potential breakdowns in the DQR reduction $A \to B$ can be circumvented by a random similarity transformation of the initial matrix. The fast DQR algorithms benefit from the Hessenberg form of B since

1. the Q factor can be computed inexpensively;

2. the computational effort in the transformation from B_k to B_{k+1} is aimed at updating only the data arrays in the upper triangular portion of B_k;

3. the stopping criterion, the shifting strategies and the deflation techniques implemented in the LAPACK routines for general Hessenberg matrices can be incorporated in the fast DQR eigenvalue algorithms;

4. potential breakdowns in the iterative phase can simply be ruled out by using exceptional shifts.

The analysis of the computational and numerical properties of the resulting composite DQR algorithms is planned for future work.

6. REFERENCES

[1] D. A. Bini, F. Daddi, and L. Gemignani. On the shifted QR iteration applied to companion matrices. *Electron. Trans. Numer. Anal.*, 18:137–152 (electronic), 2004.

[2] D. A. Bini, Y. Eidelman, L. Gemignani, and I. Gohberg. Fast QR eigenvalue algorithms for Hessenberg matrices which are rank-one perturbations of unitary matrices. Technical Report 1587, Dipartimento di Matematica, Università di Pisa, 2005. In press in *SIAM J. Matrix Anal. Appl.*.

[3] D. A. Bini, Y. Eidelman, L. Gemignani, and I. Gohberg. The unitary completion and QR iterations for a class of structured matrices. In press in *Mathematics of Computation*.

[4] D. A. Bini and G. Fiorentino. Design, analysis, and implementation of a multiprecision polynomial rootfinder. *Numer. Algorithms*, 23(2-3):127–173, 2000.

[5] D. A. Bini, L. Gemignani, and V. Y. Pan. Improved initialization of the accelerated and robust QR-like polynomial root-finding. *Electron. Trans. Numer. Anal.*, 17:195–205 (electronic), 2004.

[6] D. A. Bini, L. Gemignani, and V. Y. Pan. Fast and stable QR eigenvalue algorithms for generalized companion matrices and secular equations. *Numer. Math.*, 100(3):373–408, 2005.

[7] C. Carstensen. Linear construction of companion matrices. *Linear Algebra and Appl.*, 149:191–214, 1991.

[8] S. Chandrasekaran, M. Gu, J. Xia, and J. Zhu. A fast QR algorithm for companion matrices. 2006.

[9] J. Della-Dora. Numerical linear algorithms and group theory. *Linear Algebra and Appl.*, 10:267–283, 1975.

[10] S. Delvaux and M. Van Barel. Unitary rank structured matrices. Technical Report TW464, Department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200A, 3000 Leuven (Heverlee), Belgium, July 2006.

[11] P. Dewilde and A. J. van der Veen. *Time-varying systems and computations*. Kluwer Academic Publishers, Boston, MA, 1998.

[12] Y. Eidelman, L. Gemignani, and I. Gohberg. On the fast reduction of a quasiseparable matrix to Hessenberg and tridiagonal forms. *Linear Algebra Appl.*, 420, 2007.

[13] Y. Eidelman and I. Gohberg. On a new class of structured matrices. *Integral Equations Operator Theory*, 34:293–324, 1999.

[14] Y. Eidelman and I. Gohberg. Fast inversion algorithms for a class of structured operator matrices. *Linear Algebra Appl.*, 371:153–190, 2003.

[15] Y. Eidelman, I. Gohberg, and V. Olshevsky. The QR iteration method for Hermitian quasiseparable matrices of an arbitrary order. *Linear Algebra Appl.*, 404:305–324, 2005.

[16] L. Elsner. On some algebraic problems in connection with general eigenvalue algorithms. *Linear Algebra Appl.*, 26:123–138, 1979.

[17] D. Fasino. Rational Krylov matrices and QR steps on Hermitian diagonal-plus-semiseparable matrices. *Numer. Linear Algebra Appl.*, 12(8):743–754, 2005.

[18] S. Fortune. Polynomial root finding using iterated eigenvalue computation. In *Proceedings of the 2001 International Symposium on Symbolic and Algebraic Computation*, pages 121–128 (electronic), New York, 2001. ACM.

[19] S. Fortune. An iterated eigenvalue algorithm for approximating roots of univariate polynomials. *J. Symbolic Comput.*, 33(5):627–646, 2002. Computer algebra (London, ON, 2001).

[20] L. Gemignani. Quasiseparable structures of companion pencils under the QZ-algorithm. *Calcolo*, 42(3-4):215–226, 2005.

[21] G. I. Hargreaves. *Topics in matrix computations: stability and efficiency of algorithms*. PhD thesis, School of Mathematics, University of Manchester, 2005.

[22] B. T. Smith. Error bounds for zeros of a polynomial based upon Gerschgorin's theorems. *J. Assoc. Comput. Mach.*, 17:661–674, 1970.

[23] F. Uhlig. The DQR algorithm, basic theory, convergence, and conditional stability. *Numer. Math.*, 76(4):515–553, 1997.

[24] R. Vandebril, M. Van Barel, and N. Mastronardi. An implicit *QR* algorithm for symmetric semiseparable matrices. *Numer. Linear Algebra Appl.*, 12(7):625–658, 2005.