Quotient-difference type generalizations of the power method and their analysis

Avram SIDI
Computer Science Department, Technion — Israel Institute of Technology, Haifa 32000, Israel

William F. FORD
Computer Services Division, NASA — Lewis Research Center, Cleveland, OH 44135, USA

Received 24 September 1989
Revised 15 March 1990

Abstract: The recursion relations that were proposed by Ford and Sidi (1988) for implementing vector extrapolation methods are used for devising generalizations of the power method for linear operators. These generalizations are shown to produce approximations to largest eigenvalues of a linear operator under certain conditions. They are similar in form to the quotient-difference algorithm and share similar convergence properties with the latter. These convergence properties resemble also those obtained for the basic LR and QR algorithms. Finally, it is shown that the convergence rate produced by one of these generalizations is twice as fast for normal operators as it is for nonnormal operators.

Keywords: Vector extrapolation methods, power method, quotient-difference algorithm.

1. Introduction

Let $\boldsymbol{B}$ be a normed linear space over the field of complex numbers, and denote the norm associated with $\boldsymbol{B}$ by $\|\cdot\|$. In case $\boldsymbol{B}$ is also an inner product space, we adopt the following convention for the homogeneity property of the inner product: for y, $z \in \boldsymbol{B}$ and α, β complex numbers, the inner product $(\cdot, \cdot)$ satisfies $(\alpha y, \beta z) = \bar{\alpha}\beta(y, z)$. The norm in this case is the one induced by the inner product, i.e., if $x \in \boldsymbol{B}$, $\|x\| = \sqrt{(x, x)}$.

Let x_i, $i = 0, 1, \ldots$, be a sequence in $\boldsymbol{B}$. We shall assume that

$$x_m \sim \sum_{i=1}^{\infty} \nu_i \lambda_i^m \quad \text{as } m \to \infty. \tag{1.1}$$

Here $\nu_1, \nu_2, \ldots$ are linearly independent vectors in $\boldsymbol{B}$. λ_i are distinct scalars ordered such that

$$|\lambda_1| \geqslant |\lambda_2| \geqslant |\lambda_3| \geqslant \cdots, \tag{1.2}$$

and satisfy

$$\lambda_i \neq 0, \quad i = 1, 2, \ldots, \qquad \lambda_i \neq \lambda_j \quad \text{if } i \neq j. \tag{1.3}$$

Also there can be only a finite number of λ_i's having the same modulus. The meaning of (1.1) is

0377-0427/90/$03.50 © 1990 - Elsevier Science Publishers B.V. (North-Holland)

that for any positive integer N there exist a positive constant K and a positive integer m_0 that depend only on N, such that for every $m \geqslant m_0$,

$$\left\| x_m - \sum_{i=1}^{N-1} \nu_i \lambda_i^m \right\| \leqslant K \,|\lambda_N|^m. \tag{1.4}$$

Sequences of the form described above arise naturally, for example, in the matrix iterative process

$$x_{j+1} = A x_j, \quad j = 0, 1, \ldots, \tag{1.5}$$

in which x_0 is an arbitrary column vector, and A is a square matrix whose nonzero eigenvalues have only associated eigenvectors and no principal vectors. In this case, λ_1, $\lambda_2, \ldots$ are some or all of the nonzero eigenvalues of A, depending on the spectral decomposition of x_0, and ν_1, $\nu_2, \ldots$ are proportional to its corresponding eigenvectors. Also the infinite sum $\Sigma_{i=1}^{\infty} \nu_i \lambda_i^m$ in (1.1) is now finite, and the asymptotic relation there becomes an equality.

In the next section we present three recursive techniques, which we designate the qd-MMPE, qd-TEA and qd-MPE algorithms, that are based solely on the vector sequence x_0, x_1, $x_2, \ldots$, and that can be used for obtaining approximations to λ_1, λ_2, $\lambda_3, \ldots$, in this order. These techniques generalize the well-known power method and resemble the quotient-difference (qd) algorithm. In Section 3 we analyze the convergence properties of these techniques and show that they indeed behave like the qd algorithm. We recall that the qd algorithm is used in approximating the poles of meromorphic functions in general, and the zeros of polynomials in particular. For detailed descriptions of the qd algorithm, see, for example, [3, Chapter 7] and [4, Chapter 3]. The convergence rates derived for the methods of the present work are also very similar to those derived for the basic LR and QR algorithms, see, for example, [5].

In Section 4 we will analyze one of the recursive techniques of Section 2, namely the qd-MPE algorithm, in conjunction with vector sequences x_m, $m = 0, 1, \ldots$, that are generated by iterating with a normal operator, and we will show that the rate of convergence for this case is twice that obtained for an arbitrary nonnormal operator that has the same spectrum.

Finally, we mention that the recursive techniques developed in this work are based on the recursive algorithms that were developed in [2] for implementing some vector extrapolation methods, such as the modified minimal polynomial extrapolation (MMPE), the topological epsilon algorithm (TEA), and the minimal polynomial extrapolation (MPE), from which their names are derived. The methods of proof relating to the convergence of these techniques are similar to those that were used in [6,8] in the analysis of the above-mentioned vector extrapolation methods, and in [7] in the analysis of some other recent extensions of the power method as they are applied to normal operators. We also mention that the scalar and vector epsilon algorithms of [9,10] can also be used as generalizations of the power method and produce results similar to the ones given in Section 3 of the present work. This has been shown in [1].

2. Development of the algorithms

In this section we follow very closely the developments and notation of [2].

In all three methods that we develop in this section we assume that the vector sequence x_0, x_1, $x_2, \ldots$, is given. We also assume that a sequence of scalar quantities μ_n^m, m, $n \geqslant 0$, is given.

These quantities are determined in different forms from the vectors x_i. Their exact definition will be given later in this section. In addition, we denote by b an arbitrary sequence $b_0, b_1, b_2, \ldots$. Thus μ^j stands for the sequence $\mu_0^j, \mu_1^j, \mu_2^j, \ldots$.

We define $G_k^{n,m}$ to be the determinant

$$G_k^{n,m} = \begin{vmatrix} \mu_n^m & \mu_{n+1}^m & \cdots & \mu_{n+k-1}^m \\ \mu_n^{m+1} & \mu_{n+1}^{m+1} & \cdots & \mu_{n+k-1}^{m+1} \\ \vdots & \vdots & & \vdots \\ \mu_n^{m+k-1} & \mu_{n+1}^{m+k-1} & \cdots & \mu_{n+k-1}^{m+k-1} \end{vmatrix}, \tag{2.1}$$

and $f_k^{n,m}(b)$ to be the determinant

$$f_k^{n,m}(b) = \begin{vmatrix} b_n & b_{n+1} & \cdots & b_{n+k} \\ \mu_n^m & \mu_{n+1}^m & \cdots & \mu_{n+k}^m \\ \vdots & \vdots & & \vdots \\ \mu_n^{m+k-1} & \mu_{n+1}^{m+k-1} & \cdots & \mu_{n+k}^{m+k-1} \end{vmatrix}. \tag{2.2}$$

We also define

$$G_0^{n,m} = 1 \quad \text{and} \quad f_0^{n,m}(b) = b_n. \tag{2.3}$$

Finally, we let

$$T_k^{n,m}(b) = \frac{f_k^{n,m}(b)}{G_k^{n+1,m}}. \tag{2.4}$$

Note that by (2.3) and (2.4),

$$T_0^{n,m}(b) = b_n. \tag{2.5}$$

2.1. The qd-MMPE algorithm

Let $Q_1, Q_2, \ldots$ be a sequence of linearly independent bounded linear functionals on $\boldsymbol{B}$, and let

$$\mu_n^m = Q_{m+1}(x_n), \quad m, n \geqslant 0. \tag{2.6}$$

We recall that in case $\boldsymbol{B}$ is a complete inner product space, each Q_j has a unique representer q_j in $\boldsymbol{B}$ in the sense that $Q_j(z) = (q_j, z)$ for all z in $\boldsymbol{B}$.

As is shown in [2, Theorem 2.1], the quantities $T_k^n(b) \equiv T_k^{n,0}(b)$ satisfy the three-term recursion relation

$$T_k^n(b) = T_{k-1}^n(b) - d_k^n T_{k-1}^{n+1}(b), \tag{2.7}$$

where

$$d_k^n = \frac{T_{k-1}^n(\mu^{k-1})}{T_{k-1}^{n+1}(\mu^{k-1})}. \tag{2.8}$$

Note that $d_1^n = \mu_n^0/\mu_{n+1}^0$ from (2.5) and (2.8). Note also that $T_k^n(\mu^j) = 0$ for $0 \leqslant j \leqslant k-1$.

As will be shown in Section 3, for the vector sequences $x_0, x_1, x_2, \ldots$ described in the beginning of Section 1, $1/d_k^n \to \lambda_k$ as $n \to \infty$ under certain conditions. Thus the developments above provide us with a technique that can produce approximations to $\lambda_1, \lambda_2, \ldots$, in this order. This technique is summarized below.

qd-MMPE algorithm

Given the sequence $x_0, x_1, x_2, \ldots$, compute μ_n^m, $m, n \geqslant 0$, by (2.6), and set $T_0^n(\mu^j) = \mu_n^j$. For $k = 1, 2, \ldots$, use (2.8) to compute d_k^n, and then use (2.7) to compute $T_k^n(\mu^j)$, $j \geqslant k$.

We note that d_k^n is constructed from the $k+1$ vectors $x_n, x_{n+1}, \ldots, x_{n+k}$.

Finally, we observe that $1/d_1^n = \mu_{n+1}^0/\mu_n^0 = Q_1(x_{n+1})/Q_1(x_n)$, which is exactly the approximation provided by the power method for λ_1, the largest eigenvalue in modulus of a linear operator, provided this eigenvalue is simple and satisfies $|\lambda_1| > |\lambda_2|$. In this sense the qd-MMPE algorithm generalizes the power method.

2.2. The qd-TEA algorithm

Let Q be a bounded linear functional on $\boldsymbol{B}$, and let

$$\mu_n^m = Q(x_{m+n}), \quad m, n \geqslant 0. \tag{2.9}$$

Again, in case $\boldsymbol{B}$ is a complete inner product space, Q has a unique representer q in $\boldsymbol{B}$ in the sense that $Q(z) = (q, z)$ for all z in $\boldsymbol{B}$. The rest is exactly as in the qd-MMPE algorithm. We only have to observe that d_k^n now is determined by the $2k$ vectors $x_n, x_{n+1}, \ldots, x_{n+2k-1}$. We also observe that for $k = 1$ the qd-TEA and qd-MMPE algorithms are the same when $Q = Q_1$. Consequently, as with the qd-MMPE algorithm, also with the qd-TEA algorithm the power method is obtained for $k = 1$. Hence the qd-TEA algorithm also generalizes the power method.

2.3. The qd-MPE algorithm

The scalar quantities μ_n^m are now defined by

$$\mu_n^m = (x_m, x_n), \quad m, n \geqslant 0. \tag{2.10}$$

As is shown in [2, Theorem 3.1], the quantities $T_k^n(b) \equiv T_k^{n,n}(b)$ and $\tilde{T}_k^n(b) \equiv T_k^{n,n-1}(b)$ satisfy the coupled recursion relations

$$\begin{aligned} T_k^n(b) &= T_{k-1}^n(b) - d_k^n \tilde{T}_{k-1}^{n+1}(b), \quad n \geqslant 0, \\ \tilde{T}_k^n(b) &= T_{k-1}^n(b) - \tilde{d}_k^n \tilde{T}_{k-1}^{n+1}(b), \quad n \geqslant 1, \end{aligned} \tag{2.11}$$

where

$$d_k^n = \frac{T_{k-1}^n(\mu^{n+k-1})}{\tilde{T}_{k-1}^{n+1}(\mu^{n+k-1})}, \quad n \geqslant 0, \qquad \tilde{d}_k^n = \frac{T_{k-1}^n(\mu^{n-1})}{\tilde{T}_{k-1}^{n+1}(\mu^{n-1})}, \quad n \geqslant 1. \tag{2.12}$$

In fact, there exists a very interesting four-term (lozenge) recursion relation among $T_{k+1}^n(b)$, $T_k^n(b)$, $T_k^{n+1}(b)$ and $T_{k-1}^{n+1}(b)$, see [2]. Note that $d_1^n = \mu_n^n/\mu_{n+1}^n$, $n \geqslant 0$, and $\tilde{d}_1^n = \mu_n^{n-1}/\mu_{n+1}^{n-1}$, $n \geqslant 1$, from (2.5) and (2.12). Note also that $T_k^n(\mu^j) = 0$ for $n \leqslant j \leqslant n+k-1$, and $\tilde{T}_k^n(\mu^j) = 0$ for $n-1 \leqslant j \leqslant n+k-2$.

As will be shown in Section 3, for the vector sequences x_0, x_1, $x_2, \dots$, described in the beginning of Section 1, $1/d_k^n \to \lambda_k$ and $1/\tilde{d}_k^n \to \lambda_k$ as $n \to \infty$ under certain conditions. Thus, the developments above provide us with yet another technique that can produce approximations to λ_1, $\lambda_2, \dots$, in this order. This technique is summarized below.

qd-MPE algorithm
Given the sequence x_0, x_1, $x_2, \dots$, compute μ_n^m, m, $n \geqslant 0$, by (2.10), and set $T_0^n(\mu^j) = \mu_n^j = \tilde{T}_0^n(\mu^j)$. For $k = 1, 2, \dots$, use (2.12) to compute d_k^n and $\tilde{d}_k^n$, and then use (2.11) to compute $T_k^n(\mu^j)$, $j \leqslant n-1$, $j \geqslant n+k$, and $\tilde{T}_k^n(\mu^j)$, $j \leqslant n-2$, $j \geqslant n+k-1$.

Observe that d_k^n and $\tilde{d}_k^n$ are constructed from the $k+2$ vectors x_{n-1}, x_n, $x_{n+1}, \dots, x_{n+k}$ only.

Finally, we observe that $1/d_1^n = \mu_{n+1}^n/\mu_n^n = (x_n, x_{n+1})/(x_n, x_n)$, which is exactly the Rayleigh quotient that provides an approximation to λ_1, the largest eigenvalue in modulus of a linear operator, provided this eigenvalue is simple and satisfies $|\lambda_1| > |\lambda_2|$. In this sense the qd-MPE algorithm generalizes the power method (the Rayleigh quotient). It is known that in case the vector sequence x_0, x_1, $x_2 \dots$ is generated by a normal operator, the Rayleigh quotient converges to λ_1 twice as quickly as it does for a nonnormal operator.

So far we have shown that the three algorithms above generalize the power method for a linear operator. We would now like to explain why they are of the quotient-difference type. First, the algorithms are somewhat similar in form to the quotient-difference method. Second, as is shown in the next section (see (3.20) and (3.24)), the d_k^n are all expressed in terms of four determinants the way the quantities in the quotient-difference method are. Furthermore, again as will be shown in the next section, these determinants have asymptotic expansions very similar in form to those obtained for the determinants involved in the quotient-different method. We note in passing that the determinants that appear in the quotient-difference algorithm are Hankel determinants. The same holds true for the qd-TEA algorithms as will be clear in the next section, although the indices of these determinants are different in the two methods.

3. Convergence analysis of the algorithms

We now state the main convergence results for the qd-type algorithms that were devised in the previous section.

The following assumption is common to all three methods:

$$|\lambda_{k-1}| > |\lambda_k| > |\lambda_{k+1}| \quad \text{for some integer } k \geqslant 1, \qquad \lambda_0 \equiv \infty. \tag{3.1}$$

Let us also define

$$\epsilon_k = \max\left(\left|\frac{\lambda_k}{\lambda_{k-1}}\right|, \left|\frac{\lambda_{k+1}}{\lambda_k}\right|\right), \quad k \geqslant 1. \tag{3.2}$$

Thus, $\epsilon_k < 1$.

Theorem 3.1. *In the qd-MMPE algorithm denote*

$$S_j = \begin{vmatrix} Q_1(\nu_1) & \cdots & Q_1(\nu_j) \\ \vdots & & \vdots \\ Q_j(\nu_1) & \cdots & Q_j(\nu_j) \end{vmatrix}, \quad j = 1, 2, \dots . \tag{3.3}$$

Provided (3.1) *holds, and*

$$S_{k-1} \neq 0 \quad and \quad S_k \neq 0, \tag{3.4}$$

we have

$$\frac{1}{d_k^n} = \lambda_k + O(\epsilon_k^n) \quad as\ n \to \infty. \tag{3.5}$$

Theorem 3.2. *In the qd-TEA algorithm, provided* (3.1) *holds and*

$$\prod_{i=1}^{k} Q(\nu_i) \neq 0, \tag{3.6}$$

(3.5) *is satisfied.*

Theorem 3.3. *In the qd-MPE algorithm, provided* (3.1) *holds,* (3.5) *is satisfied. Equation* (3.5) *is satisfied also when d_k^n is replaced by $\tilde{d}_k^n$.*

Note. In relation to Theorem 3.3 we realize that if the sequence $x_0, x_1, x_2, \dots$ is generated by the matrix A in the way $x_{j+1} = Ax_j$, $j = 0, 1, \dots, x_0$ being arbitrary, then the qd-MPE algorithm performs asymptotically like the qd-MMPE and qd-TEA algorithms, although it does not require additional conditions like (3.4) and (3.6).

In the proofs of Theorems 3.1–3.3 we shall make use of the following results, which are of interest in themselves.

Lemma 3.4. *Let $\sigma_1, \sigma_2, \dots$ and $\zeta_1, \zeta_2, \dots$ be two sequences of nonzero scalars, such that the σ_i are distinct, and*

$$|\zeta_1| \geqslant |\zeta_2| \geqslant \cdots . \tag{3.7}$$

Assume, furthermore, that there can be only a finite number of ζ_j's having the same modulus. Consider the determinant

$$\psi_r^n = \begin{vmatrix} u_{1,0}^{(n)} & u_{1,1}^{(n)} & \cdots & u_{1,r-1}^{(n)} \\ u_{2,0}^{(n)} & u_{2,1}^{(n)} & \cdots & u_{2,r-1}^{(n)} \\ \vdots & \vdots & & \vdots \\ u_{r,0}^{(n)} & u_{r,1}^{(n)} & \cdots & u_{r,r-1}^{(n)} \end{vmatrix}, \tag{3.8}$$

in which

$$u_{p,q}^{(n)} \sim \sum_{j=1}^{\infty} z_{j,p}\sigma_j^q \zeta_j^n \quad \text{as } n \to \infty, \tag{3.9}$$

$z_{j,p}$ *being some scalars. Define*

$$Z_{j_1,j_2,\ldots,j_r} = \begin{vmatrix} z_{j_1,1} & z_{j_1,2} & \cdots & z_{j_1,r} \\ z_{j_2,1} & z_{j_2,2} & \cdots & z_{j_2,r} \\ \vdots & \vdots & & \vdots \\ z_{j_r,1} & z_{j_r,2} & \cdots & z_{j_r,r} \end{vmatrix}. \tag{3.10}$$

Then ψ_r^n, *for* $n \to \infty$, *has the asymptotic expansion*

$$\psi_r^n \sim \sum_{1 \leqslant j_1 < j_2 < \cdots < j_r}^{\infty} Z_{j_1,j_2,\ldots,j_r} V(\sigma_{j_1}, \sigma_{j_2}, \ldots, \sigma_{j_r}) \left(\prod_{p=1}^{r} \zeta_{j_p}^n \right), \tag{3.11}$$

where $V(a_1, \ldots, a_r)$ *is the Vandermonde determinant defined by*

$$V(a_1, \ldots, a_r) = \begin{vmatrix} 1 & a_1 & \cdots & a_1^{r-1} \\ 1 & a_2 & \cdots & a_2^{r-1} \\ \vdots & \vdots & & \vdots \\ 1 & a_r & \cdots & a_r^{r-1} \end{vmatrix}. \tag{3.12}$$

(When $\sim$ *is replaced by* $=$ *in* (3.9), $\sim$ *is replaced by* $=$ *in* (3.11) *too.) If, in addition,*

$$|\zeta_r| > |\zeta_{r+1}| \quad \text{and} \quad Z_{1,2,\ldots,r} \neq 0, \tag{3.13}$$

then

$$\psi_r^n = Z_{1,2,\ldots,r} V(\sigma_1, \sigma_2, \ldots, \sigma_r) \left(\prod_{j=1}^{r} \zeta_j^n \right) \left[1 + \mathrm{O}\left(\left| \frac{\zeta_{r+1}}{\zeta_r} \right|^n \right) \right] \quad \text{as } n \to \infty. \tag{3.14}$$

Lemma 3.4 can be obtained from [7, Lemmas 2.2 and 2.3], whose technique is a generalization of that employed in [8, Theorem 3.2].

Lemma 3.5. *Let* $\zeta_1, \zeta_2, \ldots$ *be a sequence of nonzero scalars exactly as described in Lemma* 3.4. *Consider the determinant* ψ_r^n *as given in* (3.8), *but with*

$$u_{p,q}^{(n)} \sim \sum_{i=1}^{\infty} \sum_{j=1}^{\infty} z_{ij} \bar{\zeta}_i^{n+p-1} \zeta_j^{n+q} \quad \text{as } n \to \infty, \tag{3.15}$$

z_{ij} being some scalars. Define

$$Z_{j_1,\dots,j_r}^{i_1,\dots,i_r} = \begin{vmatrix} z_{i_1j_1} & z_{i_1j_2} & \cdots & z_{i_1j_r} \\ z_{i_2j_1} & z_{i_2j_2} & \cdots & z_{i_2j_r} \\ \vdots & \vdots & & \vdots \\ z_{i_rj_1} & z_{i_rj_2} & \cdots & z_{i_rj_r} \end{vmatrix}. \tag{3.16}$$

Then ψ_r^n, for $n \to \infty$, has the asymptotic expansion

$$\psi_r^n \sim \sum_{1 \leqslant i_1 < i_2 < \cdots < i_r}^{\infty} \sum_{1 \leqslant j_1 < j_2 < \cdots < j_r}^{\infty} Z_{j_1,\dots,j_r}^{i_1,\dots,i_r} \times V(\bar{\zeta}_{i_1},\dots,\bar{\zeta}_{i_r}) V(\zeta_{j_1},\dots,\zeta_{j_r}) \left(\prod_{p=1}^{r} \bar{\zeta}_{i_p}^n \right) \left(\prod_{p=1}^{r} \zeta_{j_p}^n \right). \tag{3.17}$$

(When $\sim$ is replaced by $=$ in (3.15), *$\sim$ is replaced by $=$ in* (3.17) *too.) If, in addition,*

$$|\zeta_r| > |\zeta_{r+1}| \quad \textit{and} \quad Z_{1,\dots,r}^{1,\dots,r} \neq 0, \tag{3.18}$$

then

$$\psi_r^n = Z_{1,\dots,r}^{1,\dots,r} \, | V(\zeta_1,\dots,\zeta_r) |^2 \left| \prod_{j=1}^{r} \zeta_j \right|^{2n} \left[1 + \mathrm{O}\left(\left| \frac{\zeta_{r+1}}{\zeta_r} \right|^n \right) \right] \quad \textit{as } n \to \infty. \tag{3.19}$$

Lemma 3.5 can be proved exactly as in [6, Lemma 3.2 and Theorem 3.1].

Proof of Theorem 3.1. Combining (2.8) with (2.4), (2.2) and (2.1), and recalling that $T_k^n(b) \equiv T_k^{n,0}(b)$, we obtain

$$d_k^n = \frac{G_k^{n,0} G_{k-1}^{n+2,0}}{G_k^{n+1,0} G_{k-1}^{n+1,0}}. \tag{3.20}$$

Now by (2.6) and (1.1) we have

$$\mu_n^m \sim \sum_{j=1}^{\infty} Q_{m+1}(\nu_j) \lambda_j^n \quad \text{as } n \to \infty. \tag{3.21}$$

Substituting (3.21) in the determinant expression for $G_k^{n,0}$, we see that $G_k^{n,0}$ is simply ψ_k^n of Lemma 3.4 with $u_{p,q}^{(n)} = \mu_{n+q}^{p-1}$, thus $z_{j,p} = Q_p(\nu_j)$, $\zeta_j = \lambda_j$ and $\sigma_j = \lambda_j$. Furthermore, S_j in (3.3) is $Z_{1,2,\dots,j}$ in Lemma 3.4. Therefore, Lemma 3.4 applies, and we have

$$G_k^{n,0} = S_k V(\lambda_1,\dots,\lambda_k) \left(\prod_{j=1}^{k} \lambda_j^n \right) \left[1 + \mathrm{O}\left(\left| \frac{\lambda_{k+1}}{\lambda_k} \right|^n \right) \right] \quad \text{as } n \to \infty. \tag{3.22}$$

Treating $G_k^{n+1,0}$, $G_{k-1}^{n+1,0}$ and $G_{k-1}^{n+2,0}$ similarly, and combining the results in (3.20), we obtain (3.5). □

Proof of Theorem 3.2. We observe that by (2.9) and (2.1)

$$\mu_n^m \sim \sum_{j=1}^{\infty} \left[Q(\nu_j)\lambda_j^m \right] \lambda_j^n \quad \text{as } n \to \infty. \tag{3.23}$$

The rest of the proof is exactly the same as that of Theorem 3.1, with $Q_{m+1}(\nu_j)$ in the latter being replaced by $Q(\nu_j)\lambda_j^m$ now. On account of this, $Z_{1,\ldots,j}$ now becomes $[\prod_{i=1}^{j} Q(\nu_j)]V(\lambda_1, \ldots, \lambda_j)$. We leave out the rest of the details. □

Here we note that $G_k^{n,0}$ for the qd-TEA algorithm is a Hankel determinant. Thus the qd-TEA algorithm resembles very strongly the qd algorithm. Furthermore, when the sequence x_0, x_1, $x_2, \ldots$ is one of scalars, then the qd-TEA algorithm can still be employed by picking the functional Q to be the identity operator, in which case $\mu_n^m = x_{m+n}$. Thus for scalar sequences satisfying all the conditions mentioned in Section 1 of this work the qd-TEA algorithm provides another form of the qd algorithm.

Proof of Theorem 3.3. Combining (2.12) with (2.4), (2.2) and (2.1), and recalling that $T_k^n(b) \equiv T_k^{n,n}(b)$ and $\tilde{T}_k^n(b) \equiv T_k^{n,n-1}(b)$, we obtain

$$d_k^n = \frac{G_k^{n,n} G_{k-1}^{n+2,n}}{G_k^{n+1,n} G_{k-1}^{n+1,n}}, \qquad \tilde{d}_k^n = \frac{G_k^{n,n-1} G_{k-1}^{n+2,n}}{G_k^{n+1,n-1} G_{k-1}^{n+1,n}}. \tag{3.24}$$

Substituting (2.10) in (2.1), and comparing with (3.8), we see that $G_k^{n,n+s} = \psi_k^n$ with $u_{p,q}^{(n)} = \mu_{n+q}^{n+s+p-1} = (x_{n+s+p-1}, x_{n+q})$. Invoking now (1.1), we see that $u_{p,q}^{(n)}$ is of the form (3.15), with $z_{ij} = (\nu_i, \nu_j)\bar{\lambda}_i^s$ and $\zeta_i = \lambda_i$. Also $Z_{1,\ldots,k}^{1,\ldots,k}$ in Lemma 3.5 now becomes

$$Z_{1,\ldots,k}^{1,\ldots,k} = \left(\prod_{i=1}^{k} \bar{\lambda}_i^s \right) \begin{vmatrix} (\nu_1, \nu_1) & \cdots & (\nu_1, \nu_k) \\ \vdots & & \vdots \\ (\nu_k, \nu_1) & \cdots & (\nu_k, \nu_k) \end{vmatrix} \equiv \left(\prod_{i=1}^{k} \bar{\lambda}_i^s \right) U_{1,\ldots,k}. \tag{3.25}$$

Since the vectors ν_1, $\nu_2, \ldots$ in (1.1) are assumed to be linearly independent, the Gram determinant $U_{1,\ldots,k}$ of the vectors $\nu_1, \ldots, \nu_k$ is positive. Thus $Z_{1,\ldots,k}^{1,\ldots,k} \neq 0$. Consequently, Lemma 3.5 applies, and we obtain

$$G_k^{n,n+s} = \left(\prod_{i=1}^{k} \bar{\lambda}_i^s \right) U_{1,\ldots,k} \, | V(\lambda_1, \ldots, \lambda_k) |^2 \left| \prod_{j=1}^{k} \lambda_j \right|^{2n} \left[1 + \mathrm{O}\left(\left| \frac{\lambda_{k+1}}{\lambda_k} \right|^n \right) \right]$$
$$\text{as } n \to \infty. \tag{3.26}$$

Invoking (3.26) in (3.24), we obtain the required result. □

4. The qd-MPE algorithm for normal operators

As we have shown in Theorem 3.3, the convergence rate of the qd-MPE algorithm is the same as those of the qd-MMPE and qd-TEA algorithms, in general. When the vectors ν_1, $\nu_2, \ldots$ in the expansion (1.1) are orthogonal with respect to the inner product $(\cdot, \cdot)$, however, the convergence rate of the qd-MPE method becomes twice that given in Theorem 3.3, as will be shown shortly in Theorem 4.1. When the vector sequence $x_0, x_1, \ldots$ is generated by the matrix iterative process $x_{j+1} = Ax_j$, $j = 0, 1, \ldots$, with x_0 arbitrary, then, as we mentioned in the Introduction, x_m satisfies (1.1), with λ_i there being nonzero eigenvalues of A that possess only eigenvectors and no principal vectors, and v_i there being proportional to the corresponding eigenvectors. Now, when A is a normal operator, it has only eigenvectors, and these form an orthogonal set with respect to the inner product. We recall that Hermitian and anti-Hermitian operators are normal. Thus Theorem 4.1 applies to vector sequences generated by matrix iterative processes with normal matrices and with the Euclidean inner product.

Theorem 4.1. *In the qd-MPE algorithm, provided* (3.1) *holds and the vectors* $v_1, v_2, \ldots$ *in* (1.1) *satisfy*

$$(v_i, v_j) = z_j \delta_{ij}, \quad (\textit{thus } z_j > 0), \tag{4.1}$$

we have

$$\frac{1}{d_k^n} = \lambda_k + O(\epsilon_k^{2n}) \quad \textit{as } n \to \infty. \tag{4.2}$$

When $\lambda_j > 0$, $j = 1, 2, \ldots$, (4.2) *can be refined considerably. (In this case, the condition in* (3.1) *is automatically satisfied.) If* $\lambda_{k+1}/\lambda_k > \lambda_k/\lambda_{k-1}$, *then*

$$\frac{1}{d_k^n} \sim \lambda_k - \beta_k \left(\frac{\lambda_{k+1}}{\lambda_k} \right)^{2n} \quad \textit{as } n \to \infty, \ \textit{some } \beta_k > 0, \tag{4.3}$$

i.e., *the sequence* $\{1/d_k^n\}_{n=0}^{\infty}$ *converges to* λ_k *monotonically from below, and if* $\lambda_{k+1}/\lambda_k < \lambda_k/\lambda_{k-1}$, *then*

$$\frac{1}{d_k^n} \sim \lambda_k + \gamma_k \left(\frac{\lambda_k}{\lambda_{k-1}} \right)^{2n} \quad \textit{as } n \to \infty, \ \textit{some } \gamma_k > 0, \tag{4.4}$$

i.e., *the sequence* $\{1/d_k^n\}_{n=0}^{\infty}$ *converges to* λ_k *monotonically from above.* β_k *and* γ_k *are given in the proof below.*

All the results above hold when d_k^n *is replaced by* $\tilde{d}_k^n$, *though with different* β_k *and* γ_k.

Proof. We saw in the proof of Theorem 3.3 that $G_k^{n,n+s} = \psi_k^n$ with $u_{p,q}^{(n)} = (x_{n+s+p-1}, x_{n+q})$. Invoking now (1.1) and (4.1), we see that $u_{p,q}^{(n)}$ is of the form (3.9), $z_{j,p} = z_j \bar{\lambda}_j^{s+p-1}$, $\sigma_j = \lambda_j$ and $\zeta_j = |\lambda_j|^2$. Also $Z_{j_1, \ldots, j_k}$ in Lemma 3.4 becomes

$$Z_{j_1, \ldots, j_k} = \left(\prod_{p=1}^{k} z_{j_p} \right) \left(\prod_{p=1}^{k} \bar{\lambda}_{j_p}^{s} \right) V(\bar{\lambda}_{j_1}, \ldots, \bar{\lambda}_{j_k}). \tag{4.5}$$

Substituting all this in (3.11), we obtain for $n \to \infty$

$$G_k^{n,n+s} \sim \sum_{1 \leqslant j_1 < \cdots < j_k} \left(\prod_{p=1}^{k} \bar{\lambda}_{j_p}^{s} \right) \left(\prod_{p=1}^{k} z_{j_p} \right) |V(\lambda_{j_1}, \ldots, \lambda_{j_k})|^2 \left| \prod_{p=1}^{k} \lambda_{j_p} \right|^{2n}. \tag{4.6}$$

Consequently, all the conditions of Lemma 3.4 are satisfied, and

$$G_k^{n,n+s} = \left(\prod_{j=1}^{k} \bar{\lambda}_j^s\right)\left(\prod_{j=1}^{k} z_j\right) |V(\lambda_1,\dots,\lambda_k)|^2 \left|\prod_{j=1}^{k} \lambda_j\right|^{2n} \left[1 + \mathrm{O}\left(\left|\frac{\lambda_{k+1}}{\lambda_k}\right|^{2n}\right)\right]$$
$$\text{as } n \to \infty. \tag{4.7}$$

Invoking (4.7) in (3.24), (4.2) follows.

Now when the λ_j's are all real, from (4.6) and (4.1), we see that $G_k^{n,n+s}$, $s = 0, \pm 2, \pm 4, \dots$, as $n \to \infty$, are all real and positive, and when the λ_j's are all positive, $G_k^{n,n+s}$ are positive for all s. If the vectors x_j are generated by the matrix iterative process given in (1.5), then this holds for all n, and not only as $n \to \infty$. In case all λ_j's are positive, (1.2) and (3.1) become

$$\lambda_1 > \lambda_2 > \cdots > \lambda_{k-1} > \lambda_k > \lambda_{k+1} > \cdots, \tag{4.8}$$

so that (3.1) is automatically satisfied now. The two most dominant terms in the summation in (4.6) are those with the indices $j_1, \dots, j_k = 1, 2, \dots, k$ and $j_1, \dots, j_k = 1, 2, \dots, k-1, k+1$. Thus

$$G_k^{n,n+s} = \left(\prod_{j=1}^{k} z_j\right) |V(\lambda_1,\dots,\lambda_k)|^2 \left(\prod_{j=1}^{k} \lambda_j^{2n+s}\right)$$
$$\times \left[1 + \alpha_k\left(\frac{\lambda_{k+1}}{\lambda_k}\right)^{2n+s} + \mathrm{o}\left(\left(\frac{\lambda_{k+1}}{\lambda_k}\right)^{2n}\right)\right] \quad \text{as } n \to \infty, \tag{4.9}$$

where

$$\alpha_k = \frac{z_{k+1}}{z_k}\left[\frac{V(\lambda_1,\dots,\lambda_{k-1},\lambda_{k+1})}{V(\lambda_1,\dots,\lambda_{k-1},\lambda_k)}\right]^2 = \frac{z_{k+1}}{z_k}\prod_{j=1}^{k-1}\left(\frac{\lambda_{k+1}-\lambda_j}{\lambda_k-\lambda_j}\right)^2 > 0. \tag{4.10}$$

Consequently, for $n \to \infty$,

$$d_k^n \sim \frac{1}{\lambda_k} \frac{\left[1 + \alpha_k\left(\frac{\lambda_{k+1}}{\lambda_k}\right)^{2n}\right]\left[1 + \alpha_{k-1}\left(\frac{\lambda_k}{\lambda_{k-1}}\right)^{2n+2}\right]}{\left[1 + \alpha_k\left(\frac{\lambda_{k+1}}{\lambda_k}\right)^{2n+1}\right]\left[1 + \alpha_{k-1}\left(\frac{\lambda_k}{\lambda_{k-1}}\right)^{2n+1}\right]}$$
$$\sim \frac{1}{\lambda_k}\left[1 + \alpha_k\left(1 - \frac{\lambda_{k+1}}{\lambda_k}\right)\left(\frac{\lambda_{k+1}}{\lambda_k}\right)^{2n}\right]\left[1 - \alpha_{k-1}\left(1 - \frac{\lambda_k}{\lambda_{k-1}}\right)\left(\frac{\lambda_k}{\lambda_{k-1}}\right)^{2n+1}\right]$$
$$\sim \frac{1}{\lambda_k}\left[1 + \alpha_k\left(1 - \frac{\lambda_{k+1}}{\lambda_k}\right)\left(\frac{\lambda_{k+1}}{\lambda_k}\right)^{2n} - \alpha_{k-1}\left(1 - \frac{\lambda_k}{\lambda_{k-1}}\right)\left(\frac{\lambda_k}{\lambda_{k-1}}\right)^{2n+1}\right]. \tag{4.11}$$

The rest of the proof can now be completed easily. We only note that

$$\beta_k = \alpha_k\left(1 - \frac{\lambda_{k+1}}{\lambda_k}\right)$$

in (4.3), and

$$\gamma_k = \alpha_{k-1}\left(1 - \frac{\lambda_k}{\lambda_{k-1}}\right)\frac{\lambda_k}{\lambda_{k-1}}$$

in (4.4). □

References

[1] C. Brezinski, Computation of the eigenelements of a matrix by the ϵ-algorithm, *Linear Algebra Appl.* **11** (1975) 7–20.

[2] W.F. Ford and A. Sidi, Recursive algorithms for vector extrapolation methods, *Appl. Numer. Math.* **4** (6) (1988) 477–489.

[3] P. Henrici, *Applied and Computational Complex Analysis, Vol. 1* (Wiley, New York, 1974).

[4] A.S. Householder, *The Numerical Treatment of a Single Nonlinear Equation* (McGraw-Hill, New York, 1970).

[5] B.N. Parlett, The LU and QR algorithms, in: A. Ralston and H.S. Wilf, Eds., *Mathematical Methods for Digital Computers, Vol. II* (Wiley, New York, 1967) 116–130.

[6] A. Sidi, Convergence and stability properties of minimal polynomial and reduced rank extrapolation algorithms, *SIAM J. Numer. Anal.* **23** (1986) 197–209.

[7] A. Sidi, On extensions of the power method for normal operators, *Linear Algebra Appl.* **120** (1989) 207–224.

[8] A. Sidi, W.F. Ford and D.A. Smith, Acceleration of convergence of vector sequences, *SIAM J. Numer. Anal.* **23** (1986) 178–196.

[9] P. Wynn, On a device for computing the $e_m(S_n)$ transformation, *MTAC* **10** (1956) 91–96.

[10] P. Wynn, Acceleration techniques for iterated vector and matrix problems, *Math. Comp.* **16** (1962) 301–322.