



This is to certify that the

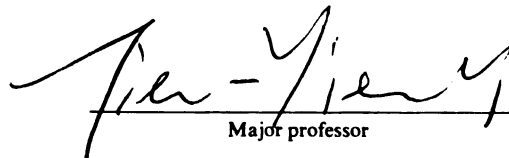
dissertation entitled
**QUASI-LAGUERRE ITERATION AND ITS
APPLICATION IN SOLVING SYMMETRIC
TRIDIAGONAL EIGENVALUE PROBLEMS**

presented by

MING JIN

has been accepted towards fulfillment
of the requirements for

PH.D. degree in MATHEMATICS



Major professor

Date JULY 20, 1995

LIBRARY
Michigan State
University

PLACE IN RETURN BOX to remove this checkout from your record.
TO AVOID FINES return on or before date due.

DATE DUE	DATE DUE	DATE DUE
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____

MSU is An Affirmative Action/Equal Opportunity Institution

c:\circ\datedue.pm3-p.1

**QUASI-LAGUERRE ITERATION AND ITS
APPLICATION IN SOLVING SYMMETRIC
TRIDIAGONAL EIGENVALUE PROBLEMS**

By

Ming Jin

A DISSERTATION

**Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of**

DOCTOR OF PHILOSOPHY

Department of Mathematics

1995

ABSTRACT

QUASI-LAGUERRE ITERATION AND ITS APPLICATION IN SOLVING SYMMETRIC TRIDIAGONAL EIGENVALUE PROBLEMS

By

Ming Jin

In this work, the quasi-Laguerre iteration is established in the spirit of Laguerre's iteration for solving polynomial f with all real zeros. The new iteration, which maintains the monotonicity and global convergence of the Laguerre iteration, no longer needs to evaluate f'' . The ultimate convergence rate is $\sqrt{2} + 1$.

Based on the quasi-Laguerre iteration, an algorithm is developed and tested for solving the eigenvalue problem of symmetric tridiagonal matrices. In general, it improves the speed of Laguerre's iteration substantially and the accuracy is at least as good as any standard algorithm for the symmetric tridiagonal eigenproblem. More preciously, the algorithm, like the Laguerre iteration, enjoys the flexibility in evaluating partial spectrum.

Although the quasi-Laguerre iteration is derived for solving polynomials with only real zeros, the iteration itself can be extended to solve polynomials with complex zeros and the local convergence can still be guaranteed with the same convergence rate $\sqrt{2} + 1$ as the real case.

To my parents

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my thesis advisor, Professor *Tien-Yien Li*, for his guidance through the years I have spent at Michigan State University. His encouragement, helpful directions and lively discussions with me make this work possible.

I would like to thank my dissertation committee members Professor *Chichia Chiu*, Professor *Richard Hill*, Professor *William T. Sledd* and Professor *Zhengfang Zhou* for their valuable suggestions and their time.

I would also like to thank Dr. *Zhonggang Zeng* for his precious advice and collaboration.

Contents

0	Introduction	1
1	Derivation of the quasi-Laguerre iteration	5
2	The formulation and convergence	13
2.1	The formulation	13
2.2	The convergence properties	15
3	On symmetric tridiagonal eigenvalue problems	26
3.1	Evaluation of the logarithmic derivative f'/f of the de- terminant	26
3.2	The split-merge process	29
3.3	Cluster and deflation	32
3.4	Partial spectrum	33
3.5	Practical considerations in implementation	34
3.5.1	Stopping criteria	34
3.5.2	Initial points for the quasi-Laguerre iteration . . .	35
3.5.3	An alternative approach for clusters of eigenvalues	39
3.6	Numerical tests	42
3.6.1	Testing matrices	45
3.6.2	Speed test in evaluating eigenvalues without com- puting eigenvectors	46
3.6.3	Speed test in evaluating eigenpairs	49
3.6.4	Accuracy test	52
4	The complex quasi-Laguerre iteration	54
4.1	The formulation	54
4.2	Convergence properties	56

List of Tables

1	Accelerated quasi-Laguerre iteration	44
2	Execution time for evaluating eigenpairs	51
3	Accuracy test of various algorithms	53

List of Figures

1	Algorithm DETEVL	28
2	Split and merge processes	31
3	Cluster of eigenvalues	32
4	The sign pattern of $-f'/f$	36
5	Algorithm INIPts for the initial points	38
6	Algorithm ACCQ-LAG for acceleration	43
7	Execution time in evaluating eigenvalues only (I)	47
8	Execution time in evaluating eigenvalues only (II)	48

0 Introduction

The globally and monotonically convergent Laguerre's iteration

$$L_{\pm}(x) = x + \frac{n}{\left(-\frac{f'(x)}{f(x)}\right) \pm \sqrt{(n-1) \left[(n-1) \left(-\frac{f'(x)}{f(x)}\right)^2 - n \left(\frac{f''(x)}{f(x)}\right) \right]}} \quad (1)$$

has been successfully used to find the eigenvalues of an $n \times n$ symmetric tridiagonal matrix T with nonzero sub-diagonal entries by approximating the zeros, always real and simple, of its characteristic polynomial

$$f(\lambda) = \det[T - \lambda I]. \quad (2)$$

Remarkable numerical results, in terms of both speed and accuracy, on a substantial variety of matrices have been obtained [8]. The algorithm employs the split-merge process, similar to Cuppen's divide and conquer strategy, which provides an excellent set of starting values making the algorithm naturally parallel. Impressive speedups of the parallel version of the algorithm were reported in [13]. For $2,000 \times 2,000$ random matrices, the algorithm can reach a speedup of 52 on an N -cube with 64 nodes.

Laguerre's iteration, or modified Laguerre's iteration when clusters occur, converges ultimately with a very fast cubic convergence rate. Nonetheless, the most important advantage of Laguerre's iteration in solving (2), in contrast to Newton's iteration, is its monotone convergence. However, comparing with Newton's iteration, a major disadvantage of Laguerre's iteration is the evaluation of f'' , which is relatively time consuming. In this work, a new algorithm is designed in the spirit of Laguerre's iteration for finding zeros of (2). The algorithm, without evaluating f'' , maintains the monotonicity of Laguerre's iteration and

converges globally with ultimate convergence rate $\sqrt{2} + 1$ in finding zeros of any polynomial with all real zeros.

The formula (1) of Laguerre's iteration can be derived in diverse ways. The best one seems to be to answer the following question [7]:

QUESTION 1: Among all polynomials $p(x)$ of degree n with n real zeros and with $p(x_0) = f(x_0) \neq 0$, $p'(x_0) = f'(x_0)$ and $p''(x_0) = f''(x_0)$ at a specified real x_0 , which one has a zero closest to x_0 ? and where?

In general, $L_+(x_0)$ in (1) gives the closest zero from the right of all those polynomials and $L_-(x_0)$ gives the closest one from the left.

To avoid the evaluation of f'' , we revise the above optimization problem as follows:

QUESTION 2: Given two specified reals $a < b$, among all polynomials $p(x)$ of degree n with n real zeros, none of which lie in $[a, b]$, and with

$$\frac{p'(a)}{p(a)} = \frac{f'(a)}{f(a)} \quad \text{and} \quad \frac{p'(b)}{p(b)} = \frac{f'(b)}{f(b)}, \quad (3)$$

which one has a zero closest to a from the right or from the left? and where?

The optimization problem for this class of polynomials can be easily solved and the solution has a closed form. To convert the solution of this optimization problem to an iteration scheme, which we call the *quasi-Laguerre iteration*, with the assumption that $x^{(k-1)}$ and $x^{(k)}$ are available, we may

- (i) let $a = x^{(k-1)}$, $b = x^{(k)}$ and let $x^{(k+1)}$ be the closest zero to $x^{(k)}$ from the right when $x^{(k-1)} < x^{(k)}$ and a zero of f is bigger than $x^{(k)}$,

- (ii) let $a = x^{(k)}$, $b = x^{(k-1)}$ and let $x^{(k+1)}$ be the closest zero to $x^{(k)}$ from the left when $x^{(k-1)} > x^{(k)}$ and a zero of f is less than $x^{(k)}$.

In this manner, the sequence generated will converge, in fact, monotonically to a zero of f .

The optimization problem in Question 2 is formulated in a more general form in Chapter 1 to account for multiple zeros. The answer to this problem leads to the quasi-Laguerre iteration. In Section 2.1, the iteration is formulated. The ultimate rate of convergence of our iteration so obtained is $\sqrt{2} + 1$ which will be shown in Section 2.2.

One of our main goals is to use the newly derived quasi-Laguerre iteration for approximating eigenvalues of symmetric tridiagonal matrices. The details of this implementation are described in Chapter 3, including some practical considerations in implementation (Section 3.5) and comprehensive numerical results on diverse types of matrices (Section 3.6). Apparently, our algorithm improves the speed of Laguerre's iteration considerably in general. Like Laguerre's iteration, our algorithm is inherently parallel and has a great capacity for vectorization. An intensive experiment in this regard will be reported in a future work.

Although the quasi-Laguerre iteration is derived for solving polynomials with only real zeros, the iteration itself can be extended to solve polynomials with complex zeros in the same way the Laguerre iteration is extended. In Chapter 4, we will first define and formulate the complex quasi-Laguerre iteration. Then in Section 4.2, we will show that the local convergence of our iteration is guaranteed for simple zeros and the ultimate convergence rate is $\sqrt{2} + 1$, similar to the real case. Comparing with Newton's and Laguerre's iteration, the most attractive aspect of our quasi-Laguerre iteration in complex case is its fast convergence as well as the elimination of evaluating the second derivative,

even though it no longer enjoys the monotone or global convergence.

1 Derivation of the quasi-Laguerre iteration

Let f be a polynomial of degree n with all of its zeros being real. For $a < b$ with $f(a)f(b) \neq 0$, let $\mathcal{F}$ be the class of polynomials $p(x)$ of degree n that satisfy the following conditions:

- (i) all zeros of $p(x)$ are real;
- (ii) none of the zeros of $p(x)$ are in $[a, b]$;
- (iii) $p(a)p(b) \neq 0$ and

$$\frac{p'(a)}{p(a)} = \frac{f'(a)}{f(a)} \equiv q(a), \quad \frac{p'(b)}{p(b)} = \frac{f'(b)}{f(b)} \equiv q(b). \quad (4)$$

We pose an optimization problem more general than Question 2:

QUESTION 3: For $p \in \mathcal{F}$, consider its m -th ($m < n$) zero to the right (or to the left) of a . Which polynomial in $\mathcal{F}$ has the closest one to a from the right (or from the left)? and where?

To answer this question, let $z_1, z_2, \dots, z_n$ be all the real zeros of a given polynomial $p(x)$ in $\mathcal{F}$ such that

$$z_1 \leq z_2 \leq \dots \leq z_k < a < b < z_{k+1} \leq \dots \leq z_n. \quad (5)$$

It follows from (4) that

$$\frac{p'(a)}{p(a)} = \sum_{i=1}^n \frac{1}{a - z_i} = q(a) \quad \text{and} \quad \frac{p'(b)}{p(b)} = \sum_{i=1}^n \frac{1}{b - z_i} = q(b). \quad (6)$$

For $1 \leq i \leq n$, let

$$X_i \equiv \frac{b - z_i}{a - z_i} = 1 + \frac{b - a}{a - z_i}$$

then, $X_i > 0$ for all i since no z_i falls between a and b . Moreover,

$$\begin{aligned} \sum_{i=1}^n X_i &= \sum_{i=1}^n \left(1 + \frac{b-a}{a-z_i}\right) = n + (b-a)q(a) \equiv r(a, b), \\ \sum_{i=1}^n \frac{1}{X_i} &= \sum_{i=1}^n \left(1 + \frac{a-b}{b-z_i}\right) = n + (a-b)q(b) = r(b, a), \end{aligned} \quad (7)$$

and from (5),

$$X_k \geq X_{k-1} \geq \cdots \geq X_1 > 1 > X_n \geq X_{n-1} \geq \cdots \geq X_{k+1} > 0 \quad . \quad (8)$$

For convenience, we rewrite (8) as

$$W_1 \geq W_2 \geq \cdots \geq W_k > 1 > W_{k+1} \geq \cdots \geq W_n \quad (9)$$

with

$$W_l = \begin{cases} X_{k+1-l} & \text{for } l \leq k \\ X_{n+k+1-l} & \text{for } l > k. \end{cases} \quad (10)$$

From (5), the m -th zero of $p(x)$ to the left of a is z_{k-m+1} which, from (10), corresponds to W_m in (9), and minimizing $(a - z_{k-m+1})$ becomes maximizing W_m . Similarly, the m -th zero of $p(x)$ to the right of a is z_{k+m} and minimizing $(z_{k+m} - a)$ becomes minimizing W_{n-m+1} . As a convention, if $m > k$ then the m -th zero of p to the left of a will be taken as the $(n - m + 1)$ -st zero to the right of a and if $m > n - k$ then the m -th zero of p to the right of a is the $(n - m + 1)$ -st zero to the left of a .

The optimization problem in Question 3 can now be converted to the following optimizations:

(P1)

$$\begin{aligned} & \text{Max} \quad W_m \\ & \text{subject to} \quad \sum_{i=1}^n W_i = r(a, b) \\ & \quad \quad \quad \sum_{i=1}^n \frac{1}{W_i} = r(b, a) \\ & \quad \quad \quad W_1 \geq W_2 \geq \cdots \geq W_n > 0, \end{aligned} \quad (11)$$

(P2)

$$\begin{aligned}
& \text{Min} \quad W_{n-m+1} \\
& \text{subject to} \quad \sum_{i=1}^n W_i = r(a, b) \\
& \quad \quad \quad \sum_{i=1}^n \frac{1}{W_i} = r(b, a) \\
& \quad \quad \quad W_1 \geq W_2 \geq \cdots \geq W_n > 0.
\end{aligned}$$

The solution of (P1) gives the m -th closest zero to a from the left while the solution of (P2) gives the m -th closest zero to a from the right.

We first consider the optimization problem (P1).

Lemma 1.1 $r(a, b) > 0, r(b, a) > 0$ and $r(a, b)r(b, a) \geq n^2$.

Proof. By (8), we have $r(a, b) > 0$ and $r(b, a) > 0$. From the Cauchy-Schwartz inequality,

$$\begin{aligned}
n^2 &= \left(\sum_{i=1}^n \sqrt{W_i} \sqrt{\frac{1}{W_i}} \right)^2 \leq \left(\sum_{i=1}^n W_i \right) \left(\sum_{i=1}^n \frac{1}{W_i} \right) \\
&= r(a, b)r(b, a) \quad (\text{because of (11)}). \quad \square
\end{aligned}$$

Lemma 1.2 The point $\overline{W} = (\overline{W}_1, \dots, \overline{W}_n)$ satisfying (11) with $\overline{W}_1 = \overline{W}_2 = \dots = \overline{W}_m = \dots = \overline{W}_{n-1} \geq \overline{W}_n > 0$ and

$$\overline{W}_m \geq \frac{r(a, b)}{n} \tag{12}$$

is feasible to problem (P1).

Proof. From (11), $\overline{W}_m$ and $\overline{W}_n$ must satisfy

$$(n-1)\overline{W}_m + \overline{W}_n = r(a, b), \tag{13}$$

$$\frac{n-1}{\overline{W}_m} + \frac{1}{\overline{W}_n} = r(b, a). \tag{14}$$

Eliminating $\overline{W}_m$ in (13) and (14) yields

$$r(b, a)\overline{W}_n^2 - [r(a, b)r(b, a) - n^2 + 2n]\overline{W}_n + r(a, b) = 0. \quad (15)$$

By Lemma 1.1, it can be easily verified that both solutions $\overline{W}_n^{(1)}$ and $\overline{W}_n^{(2)}$ of (15) are real. In addition, they are positive since $\overline{W}_n^{(1)} + \overline{W}_n^{(2)} > 0$ and $\overline{W}_n^{(1)}\overline{W}_n^{(2)} > 0$. Let $\overline{W}_n^{(1)} \leq \overline{W}_n^{(2)}$. Since

$$\begin{aligned} \overline{W}_n^{(1)}\overline{W}_n^{(2)} &= \frac{r(a, b)}{r(b, a)} = \frac{r(a, b)}{n} \frac{n}{r(b, a)} \\ &\leq \left(\frac{r(a, b)}{n} \right)^2 \quad (\text{because of Lemma 1.1}), \end{aligned}$$

so, $\overline{W}_n^{(1)} \leq \frac{r(a, b)}{n}$. Substituting $\overline{W}_n^{(1)}$ into (13),

$$\overline{W}_m^{(1)} = \frac{1}{n-1}(r(a, b) - \overline{W}_n^{(1)}) \geq \frac{r(a, b)}{n}.$$

It follows that $\overline{W}_m^{(1)} \geq \overline{W}_n^{(1)}$, and so, $\overline{W} = (\overbrace{\overline{W}_m^{(1)}, \dots, \overline{W}_m^{(1)}}^{n-1}, \overline{W}_n^{(1)})$ is feasible. $\square$

Now, for any $W = (W_1, \dots, W_n)$ satisfying (11), we have

$$\begin{aligned} \sum_{j>m} \frac{(W_j - W_m)^2}{W_j} &\leq \sum_{j=1}^n \frac{(W_j - W_m)^2}{W_j} = \sum_{j=1}^n \left(W_j - 2W_m + \frac{W_m^2}{W_j} \right) \\ &= r(a, b) - 2nW_m + r(b, a)W_m^2. \end{aligned} \quad (16)$$

On the other hand,

$$\begin{aligned} \sum_{j>m} (W_m - W_j) &= (n-m)W_m - \sum_{j>m} W_j \\ &= (n-m)W_m - r(a, b) + \sum_{j \leq m} W_j \\ &\geq nW_m - r(a, b). \end{aligned}$$

From (12), $nW_m - r(a, b) \geq 0$ for a certain feasible point, so we may assume, in general, W_m satisfies

$$nW_m - r(a, b) \geq 0$$

since we are maximizing W_m . It follows that

$$\begin{aligned}
 [nW_m - r(a, b)]^2 &\leq \left[\sum_{j>m} (W_m - W_j) \right]^2 = \left[\sum_{j>m} \left(\frac{W_m - W_j}{\sqrt{W_j}} \cdot \sqrt{W_j} \right) \right]^2 \\
 &\leq \left(\sum_{j>m} \frac{(W_j - W_m)^2}{W_j} \right) \sum_{j>m} W_j \\
 &\quad \text{(by Cauchy-Schwartz inequality)} \\
 &\leq [r(a, b) - 2nW_m + r(b, a)W_m^2] (r(a, b) - mW_m) \\
 &\quad \text{(because of (16)).}
 \end{aligned}$$

Or,

$$mr(b, a)W_m^2 - [r(a, b)r(b, a) - n^2 + 2mn] W_m + mr(a, b) \leq 0 \quad (17)$$

and the equality holds iff $W_j = W_m$ for all $j < m$ and $W_l = W_n$ for all $l > m$.

Similarly, when solving the optimization problem (P2), one can show that, with the assumption

$$r(b, a)W_{n-m+1} - n \leq 0$$

which we may assume without loss of generality, W_{n-m+1} satisfies the same inequality, that is,

$$mr(b, a)W_{n-m+1}^2 - [r(a, b)r(b, a) - n^2 + 2mn] W_{n-m+1} + mr(a, b) \leq 0 \quad (18)$$

and the equality holds iff $W_j = W_{n-m+1}$ for all $j > n - m + 1$ and $W_l = W_1$ for all $l < n - m + 1$.

Now, taking equalities in (17) and (18) yields the quadratic equation,

$$mr(b, a)Y^2 - [r(a, b)r(b, a) - n^2 + 2mn] Y + mr(a, b) = 0.$$

Its solutions $Y_{m\pm} =$

$$\frac{r(a,b)r(b,a) - n^2 + 2mn \pm \sqrt{[r(a,b)r(b,a) - n^2][r(a,b)r(b,a) - (n-2m)^2]}}{2mr(b,a)}$$

are both real by Lemma 1.1. Furthermore, since

$$Y_{m+} + Y_{m-} = r(a,b)r(b,a) - n^2 + 2mn \geq 2mn > 0$$

and

$$Y_{m+} \cdot Y_{m-} = \frac{r(a,b)}{r(b,a)} > 0,$$

so, $Y_{m\pm} > 0$. On the other hand, it follows from (17) and (18) that

$$Y_{m-} \leq W_m \leq Y_{m+} \quad (19)$$

and

$$Y_{m-} \leq W_{n-m+1} \leq Y_{m+}. \quad (20)$$

Accordingly, $\overline{W} = (\overline{W}_1, \dots, \overline{W}_n)$ with $\overline{W}_1 = \dots = \overline{W}_m = Y_{m+}$ and $\overline{W}_{m+1} = \dots = \overline{W}_n$ is the solution of the maximization problem (P1), and, from (11),

$$\begin{aligned} mY_{m+} + (n-m)\overline{W}_n &= r(a,b), \\ \frac{m}{Y_{m+}} + \frac{n-m}{\overline{W}_n} &= r(b,a). \end{aligned}$$

Solving for $\overline{W}_n$, one can easily see that $\overline{W}_n$ is actually $Y_{(n-m)-}$. So,

$$\overline{W}_j = \begin{cases} Y_{m+} & j \leq m \\ Y_{(n-m)-} & j > m. \end{cases} \quad (21)$$

Similarly, the solution of the minimization problem (P2) is $\underline{W} = (\underline{W}_1, \dots, \underline{W}_n)$ with

$$\underline{W}_j = \begin{cases} Y_{(n-m)+} & j < n-m+1 \\ Y_{m-} & j \geq n-m+1. \end{cases} \quad (22)$$

Recall that W_j 's are defined by

$$W_j = 1 + \frac{b-a}{a-u_j} \quad j = 1, \dots, n$$

where u_j 's are zeros of a certain polynomial in $\mathcal{F}$. Thus any polynomial satisfying (21) must be in the following form

$$p_1(x) = C(x - u_{m-})^m(x - v_1)^{n-m} \quad (23)$$

where C is a constant, and

$$1 + \frac{b-a}{a-u_{m-}} = Y_{m+} \quad \text{or equivalently} \quad u_{m-} = a - \frac{b-a}{Y_{m+}-1} \quad (24)$$

and

$$1 + \frac{b-a}{a-v_1} = Y_{(n-m)-} \quad \text{or equivalently} \quad v_1 = a - \frac{b-a}{Y_{(n-m)-}-1}.$$

And any polynomial satisfying (22) must be in the form

$$p_2(x) = C(x - u_{m+})^m(x - v_2)^{n-m} \quad (25)$$

where C is a constant, and

$$1 + \frac{b-a}{a-u_{m+}} = Y_{m-} \quad \text{or equivalently} \quad u_{m+} = a - \frac{b-a}{Y_{m-}-1} \quad (26)$$

and

$$1 + \frac{b-a}{a-v_2} = Y_{(n-m)+} \quad \text{or equivalently} \quad v_2 = a - \frac{b-a}{Y_{(n-m)+}-1}.$$

From (24) and (26),

$$\frac{b-u_{m-}}{a-u_{m-}} = Y_{m+} > 0 \quad \text{and} \quad \frac{b-u_{m+}}{a-u_{m+}} = Y_{m-} > 0,$$

so both u_{m+} and u_{m-} are outside the interval (a, b) . Similarly, both v_1 and v_2 are outside the interval (a, b) since $Y_{(n-m)\pm} > 0$. Consequently, both $p_1(x)$ and $p_2(x)$ are in $\mathcal{F}$.

Moreover, from (24) and (26)

$$\begin{aligned}
 u_{m\pm} &= a - \frac{b-a}{Y_{m\mp} - 1} = a - \\
 &\quad \frac{2mr(b,a)(b-a)}{r(a,b)r(b,a)-n^2+2mn-2mr(b,a)\mp\sqrt{[r(a,b)r(b,a)-n^2][r(a,b)r(b,a)-(n-2m)^2]}} \\
 &= a + \frac{2m[n-(b-a)q(b)]}{-(b-a)R-2mq(b)\pm\sqrt{R[(b-a)^2R+4m(n-m)]}} \quad (27)
 \end{aligned}$$

where

$$q(a) = \frac{f'(a)}{f(a)}, \quad q(b) = \frac{f'(b)}{f(b)} \quad \text{and} \quad R = n \left(\frac{q(a) - q(b)}{b-a} \right) - q(a)q(b).$$

The solutions of the optimization problems in Question 3 can now be described as follows:

- (i) If f has at least m (counting multiplicities) zeros to the left of a , then the zero u_{m-} in (27) of the polynomial

$$p_1(x) = C(x - u_{m-})^m(x - v_1)^{n-m}$$

in (23) is the closest m -th zero to the left of a among all polynomials in $\mathcal{F}$. And it is clear that

$$z_{m-} \leq u_{m-} \leq \cdots \leq u_{2-} \leq u_{1-} < a < b \quad (28)$$

where z_{m-} is the m -th zero of f to the left of a .

- (ii) If f has at least m (counting multiplicities) zeros to the right of b , then the zero u_{m+} in (27) of the polynomial

$$p_2(x) = C(x - u_{m+})^m(x - v_2)^{n-m}$$

in (25) is the closest m -th zero to the right of b among all polynomials in $\mathcal{F}$. It is also clear that

$$a < b < u_{1+} \leq u_{2+} \leq \cdots \leq u_{m+} \leq z_{m+} \quad (29)$$

where z_{m+} is the m -th zero of f to the right of b .

2 The formulation and convergence

2.1 The formulation

Let f be a polynomial of degree n with all its zeros being real. Let z_M be a zero of f with multiplicity $M \geq 1$. To approximate z_M from its left, we shall use u_{m+} in (27) with $m \leq M$ to generate a monotonically increasing sequence $\{x_{m+}^{(k)}\}$ which converges to z_M as $k \rightarrow \infty$. Similarly, to approximate z_M from its right, a monotonically decreasing sequence $\{x_{m-}^{(k)}\}$ converging to z_M , as $k \rightarrow \infty$, can be generated by using u_{m-} in (27).

To be more precise, suppose z_M is on the right hand side of two starting points $x_{m+}^{(0)} < x_{m+}^{(1)}$ and none of the zeros of f lie between $x_{m+}^{(0)}$ and z_M . Then, for $k \geq 1$, we may let

$$x_{m+}^{(k+1)} = x_{m+}^{(k-1)} + \frac{2m \left[n - (x_{m+}^{(k)} - x_{m+}^{(k-1)})q(x_{m+}^{(k)}) \right]}{-(x_{m+}^{(k)} - x_{m+}^{(k-1)})R - 2mq(x_{m+}^{(k)}) + \sqrt{R \left[(x_{m+}^{(k)} - x_{m+}^{(k-1)})^2 R + 4m(n-m) \right]}} \quad (30)$$

where

$$q(x_{m+}^{(k-1)}) = \frac{f'(x_{m+}^{(k-1)})}{f(x_{m+}^{(k-1)})}, \quad q(x_{m+}^{(k)}) = \frac{f'(x_{m+}^{(k)})}{f(x_{m+}^{(k)})}$$

and

$$R = n \left(\frac{q(x_{m+}^{(k-1)}) - q(x_{m+}^{(k)})}{x_{m+}^{(k)} - x_{m+}^{(k-1)}} \right) - q(x_{m+}^{(k-1)})q(x_{m+}^{(k)}).$$

In other words, we replace a, b and u_{m+} in (27) by $x_{m+}^{(k-1)}, x_{m+}^{(k)}$ and $x_{m+}^{(k+1)}$ respectively.

Remark 2.1 Interchanging $x_{m+}^{(k-1)}$ and $x_{m+}^{(k)}$, (30) can also be written as

$$x_{m+}^{(k+1)} = x_{m+}^{(k)} +$$

$$\frac{2m \left[n - (x_{m+}^{(k-1)} - x_{m+}^{(k)})q(x_{m+}^{(k-1)}) \right]}{-(x_{m+}^{(k-1)} - x_{m+}^{(k)})R - 2mq(x_{m+}^{(k-1)}) + \sqrt{R \left[(x_{m+}^{(k-1)} - x_{m+}^{(k)})^2 R + 4m(n-m) \right]}}. \quad (31)$$

From what have been derived in the last Chapter, it is easy to see that

$$x_{m+}^{(k)} < x_{m+}^{(k+1)} \leq z_M$$

and no zeros of f fall between $x_{m+}^{(k+1)}$ and z_M . So the sequence $\{x_{m+}^{(k)}\}$ satisfies

$$x_{m+}^{(0)} < x_{m+}^{(1)} < \cdots < x_{m+}^{(k)} < \cdots \leq z_M.$$

Similarly, when two starting points $x_{m-}^{(1)} < x_{m-}^{(0)}$ are available on the right hand side of z_M with no zeros of f lie in $(z_M, x_{m-}^{(0)})$, then for $k \geq 1$, let $a = x_{m-}^{(k)}$, $b = x_{m-}^{(k-1)}$ and $u_{m-} = x_{m-}^{(k+1)}$ in (27). Namely,

$$x_{m-}^{(k+1)} = x_{m-}^{(k)} + \frac{2m \left[n - (x_{m-}^{(k-1)} - x_{m-}^{(k)})q(x_{m-}^{(k-1)}) \right]}{-(x_{m-}^{(k-1)} - x_{m-}^{(k)})R - 2mq(x_{m-}^{(k-1)}) - \sqrt{R \left[(x_{m-}^{(k-1)} - x_{m-}^{(k)})^2 R + 4m(n-m) \right]}} \quad (32)$$

where

$$q(x_{m-}^{(k)}) = \frac{f'(x_{m-}^{(k)})}{f(x_{m-}^{(k)})}, \quad q(x_{m-}^{(k-1)}) = \frac{f'(x_{m-}^{(k-1)})}{f(x_{m-}^{(k-1)})}$$

and

$$R = n \left(\frac{q(x_{m-}^{(k)}) - q(x_{m-}^{(k-1)})}{x_{m-}^{(k-1)} - x_{m-}^{(k)}} \right) - q(x_{m-}^{(k)})q(x_{m-}^{(k-1)}).$$

Again it is clear that

$$z_M \leq x_{m-}^{(k+1)} < x_{m-}^{(k)}$$

and none of the zeros of f lie between z_M and $x_{m-}^{(k+1)}$. Moreover, the sequence $\{x_{m-}^{(k)}\}$ satisfies

$$z_M \leq \cdots < x_{m-}^{(k)} < \cdots < x_{m-}^{(1)} < x_{m-}^{(0)}.$$

We call the sequences $\{x_{m\pm}^{(k)}\}$ defined by (30) (or (31)) and (32) the **quasi-Laguerre iterations**.

2.2 The convergence properties

Theorem 2.2 *Both sequences $\{x_{m\pm}^{(k)}\}$ converge to z_M monotonically.*

Proof. We will prove the theorem for the sequence $\{x_{m-}^{(k)}\}_{k=1}^{\infty}$. Similar arguments hold for the sequence $\{x_{m+}^{(k)}\}_{k=1}^{\infty}$. We shall write y_k for $x_{m-}^{(k)}$ when there is no ambiguity. Obviously, by the way it is constructed, $\{y_k\}$ is a decreasing sequence bounded below by z_M . Let $z^{(1)}, \dots, z^{(n-M)}$ be the rest of the zeros of f besides z_M . Recall that given y_{k-1} and y_k , y_{k+1} satisfies

$$\begin{aligned} \frac{m}{y_k - y_{k+1}} + \frac{n-m}{y_k - v} &= \frac{M}{y_k - z_M} + \sum_{i=1}^{n-M} \frac{1}{y_k - z^{(i)}} \\ \frac{m}{y_{k-1} - y_{k+1}} + \frac{n-m}{y_{k-1} - v} &= \frac{M}{y_{k-1} - z_M} + \sum_{i=1}^{n-M} \frac{1}{y_{k-1} - z^{(i)}} \end{aligned}$$

for a certain real number $v \notin (y_k, y_{k-1})$. Subtracting the second from the first and dividing the result by $(y_{k-1} - y_k)$ yields,

$$\begin{aligned} &\frac{m}{(y_k - y_{k+1})(y_{k-1} - y_{k+1})} + \frac{n-m}{(y_k - v)(y_{k-1} - v)} \\ &= \frac{M}{(y_k - z_M)(y_{k-1} - z_M)} + \sum_{i=1}^{n-M} \frac{1}{(y_k - z^{(i)})(y_{k-1} - z^{(i)})}. \end{aligned}$$

Each term in the above equation is positive and z_M is the closest zero to y_k , so

$$\begin{aligned} \infty &\leftarrow \frac{m}{(y_k - y_{k+1})(y_{k-1} - y_{k+1})} \quad (\text{as } k \rightarrow \infty) \\ &\leq \frac{M}{(y_k - z_M)(y_{k-1} - z_M)} + \sum_{i=1}^{n-M} \frac{1}{(y_k - z^{(i)})(y_{k-1} - z^{(i)})} \\ &< \frac{M}{(y_k - z_M)(y_{k-1} - z_M)} + \sum_{i=1}^{n-M} \frac{1}{(z_M - z^{(i)})(z_M - z^{(i)})}. \end{aligned}$$

Therefore $y_k \rightarrow z_M$. $\square$

The rate of convergence of both sequences $\{x_{m\pm}^{(k)}\}$ is given in the following theorem.

Theorem 2.3 (i) When $m < M$, the convergence of $\{x_{m\pm}^{(k)}\}$ to z_M is linear and the convergence ratio

$$\omega \equiv \lim_{k \rightarrow \infty} \frac{x_{m\pm}^{(k+1)} - z_M}{x_{m\pm}^{(k)} - z_M}$$

is the only real solution of

$$\frac{n - M - m}{n - M}x^3 - x^2 - x + \frac{M - m}{M} = 0$$

in $(0, 1)$.

(ii) When $m = M$, the convergence of $\{x_{m\pm}^{(k)}\}$ to z_M is super-linear with convergence rate $\sqrt{2} + 1$.

Proof. Again we shall prove the theorem for the sequence $\{x_{m-}^{(k)}\}_{k=1}^{\infty}$ and write y_k for $x_{m-}^{(k)}$ when there is no ambiguity. Similar arguments hold for the sequence $\{x_{m+}^{(k)}\}_{k=1}^{\infty}$.

Recall that

$$y_{k+1} = y_k - \frac{2m[n - (y_{k-1} - y_k)q(y_{k-1})]}{(y_{k-1} - y_k)R + 2mq(y_{k-1}) + \sqrt{R[(y_{k-1} - y_k)^2 R + 4m(n - m)]}} \quad (33)$$

where

$$q(y_{k-1}) = \frac{f'(y_{k-1})}{f(y_{k-1})}, \quad q(y_k) = \frac{f'(y_k)}{f(y_k)}$$

and

$$R = n \left(\frac{q(y_k) - q(y_{k-1})}{y_{k-1} - y_k} \right) - q(y_k)q(y_{k-1}).$$

Let $z^{(1)}, \dots, z^{(n-M)}$ be the rest of the zeros of f besides z_M . Then,

$$\begin{aligned} q(y_{k-1}) &= \frac{f'(y_{k-1})}{f(y_{k-1})} = \frac{M}{y_{k-1} - z_M} + Q(y_{k-1}), \\ q(y_k) &= \frac{f'(y_k)}{f(y_k)} = \frac{M}{y_k - z_M} + Q(y_k) \quad \text{with} \quad Q(y) \equiv \sum_{i=1}^{n-M} \frac{1}{y - z^{(i)}} \end{aligned}$$

and

$$\begin{aligned} R &= n \left(\frac{q(y_k) - q(y_{k-1})}{y_{k-1} - y_k} \right) - q(y_k)q(y_{k-1}) \\ &= \frac{M(n-M)}{(y_k - z_M)(y_{k-1} - z_M)} - \frac{MQ(y_{k-1})}{y_k - z_M} - \frac{MQ(y_k)}{y_{k-1} - z_M} - Q(y_k)Q(y_{k-1}) \\ &\quad + nR_1 \quad \text{with} \quad R_1 \equiv \sum_{i=1}^{n-M} \frac{1}{(y_{k-1} - z^{(i)})(y_k - z^{(i)})}. \end{aligned}$$

Since $y_k \rightarrow z_M$ as $k \rightarrow \infty$, so

$$\begin{aligned} Q(y_{k-1}) &= \sum_{i=1}^{n-M} \frac{1}{y_{k-1} - z^{(i)}} \rightarrow Q \equiv \sum_{i=1}^{n-M} \frac{1}{z_M - z^{(i)}} < \infty, \\ Q(y_k) &= \sum_{i=1}^{n-M} \frac{1}{y_k - z^{(i)}} \rightarrow Q \end{aligned}$$

and

$$R_1 = \sum_{i=1}^{n-M} \frac{1}{(y_{k-1} - z^{(i)})(y_k - z^{(i)})} \rightarrow R^* \equiv \sum_{i=1}^{n-M} \frac{1}{(z_M - z^{(i)})^2} < \infty.$$

Let $\epsilon_k = \frac{y_k - z_M}{y_{k-1} - z_M} \quad \forall k \geq 1$, then $\frac{y_{k-1} - y_k}{y_{k-1} - z_M} = 1 - \epsilon_k$. Obviously, $0 < \epsilon_k < 1$ and $0 < 1 - \epsilon_k < 1$. From (33),

$$\begin{aligned} \frac{y_{k+1} - z_M}{y_k - z_M} &= 1 - \\ &\quad \frac{2m[n - (y_{k-1} - y_k)q(y_{k-1})]}{(y_k - z_M) \left\{ (y_{k-1} - y_k)R + 2mq(y_{k-1}) + \sqrt{R[(y_{k-1} - y_k)^2 R + 4m(n-m)]} \right\}}. \end{aligned} \quad (34)$$

Now,

$$\begin{aligned} (a) \quad (y_{k-1} - y_k)q(y_{k-1}) &= (y_{k-1} - y_k) \left(\frac{M}{y_{k-1} - z_M} + Q(y_{k-1}) \right) \\ &= M(1 - \epsilon_k) + Q(\epsilon_k)(1 - \epsilon_k)(y_{k-1} - z_M) = M(1 - \epsilon_k) + O(y_{k-1} - z_M) \end{aligned}$$

$$\begin{aligned}
(b) \quad & (y_k - z_M)(y_{k-1} - z_M)R = M(n - M) - (y_{k-1} - z_M)MQ(y_{k-1}) \\
& - (y_k - z_M)MQ(y_k) - Q(y_k)Q(y_{k-1})(y_k - z_M)(y_{k-1} - z_M) \\
& + n(y_k - z_M)(y_{k-1} - z_M)R_1 = M(n - M) + O(y_{k-1} - z_M) \\
(c) \quad & (y_k - z_M)(y_{k-1} - y_k)R = (1 - \epsilon_k)(y_k - z_M)(y_{k-1} - z_M)R \\
& = (1 - \epsilon_k)M(n - M) + O(y_{k-1} - z_M) \quad (\text{because of (b) above}) \\
(d) \quad & (y_k - z_M)^2 R = \epsilon_k(y_k - z_M)(y_{k-1} - z_M)R \\
& = \epsilon_k M(n - M) + O(y_{k-1} - z_M) \quad (\text{because of (b) above}) \\
(e) \quad & (y_k - z_M)^2 R [(y_{k-1} - y_k)^2 R + 4m(n - m)] \\
& = [(y_k - z_M)(y_{k-1} - y_k)R]^2 + 4m(n - m)(y_k - z_M)^2 R \\
& = (1 - \epsilon_k)^2 M^2(n - M)^2 + 4m(n - m)\epsilon_k M(n - M) + O(y_{k-1} - z_M) \\
& \quad (\text{because of (c) and (d) above}).
\end{aligned}$$

Substituting (a), (c), (e) into (34) yields

$$\begin{aligned}
\epsilon_{k+1} &= \frac{y_{k+1} - z_M}{y_k - z_M} = 1 - \frac{2m[n - (1 - \epsilon_k)M] + O(y_{k-1} - z_M)}{M(n - M)(1 - \epsilon_k) + 2mM\epsilon_k + M(n - M)\sqrt{(1 - \epsilon_k)^2 + \frac{4m(n - m)}{M(n - M)}\epsilon_k} + O(y_{k-1} - z_M)} \\
&= G(\epsilon_k) + O(y_{k-1} - z_M) \tag{35}
\end{aligned}$$

where

$$G(\epsilon) = 1 - \frac{2m[n - (1 - \epsilon)M]}{M(n - M)(1 - \epsilon) + 2mM\epsilon + M(n - M)\sqrt{(1 - \epsilon)^2 + \frac{4m(n - m)}{M(n - M)}\epsilon}}.$$

We claim that $G(\epsilon)$ can be rewritten as

$$G(\epsilon) = \frac{2(M - m)}{M \left(1 + \epsilon + \sqrt{(1 - \epsilon)^2 + \frac{4m(n - m)}{M(n - M)}\epsilon} \right)}. \tag{36}$$

To see this, let $\Delta = \sqrt{(1 - \epsilon)^2 + \frac{4m(n - m)}{M(n - M)}\epsilon}$ and

$$A = m[n - (1 - \epsilon)M]$$

$$B = (n - M)(1 - \epsilon) + 2m\epsilon + (n - M)\Delta$$

$$C = M - m$$

$$D = (1 + \epsilon) + \Delta.$$

Then, $G(\epsilon) = 1 - \frac{2A}{MB}$ and

$$AD = m(n - M)(1 + \epsilon) + mM\epsilon(1 + \epsilon) + [m(n - M) + mM\epsilon]\Delta$$

$$BC = (M - m)(n - M)(1 - \epsilon) + 2(M - m)m\epsilon + (M - m)(n - M)\Delta.$$

So,

$$\begin{aligned} AD + BC &= mM\epsilon^2 + (M^2 - Mn + 2mn + Mm - 2m^2)\epsilon \\ &\quad + M(n - M) + [M(n - M) + mM\epsilon]\Delta. \end{aligned}$$

Meanwhile,

$$\begin{aligned} MBD &= M(n - M)(1 - \epsilon^2) + M(n - M)(1 - \epsilon)^2 + 4m(n - m)\epsilon + 2mM\epsilon \\ &\quad + 2mM\epsilon^2 + [M(n - M)(1 + \epsilon) + M(n - M)(1 - \epsilon) + 2mM\epsilon]\Delta \\ &= 2mM\epsilon^2 + (2M^2 - 2Mn + 4mn + 2Mm - 4m^2)\epsilon \\ &\quad + 2M(n - M) + [2M(n - M) + 2mM\epsilon]\Delta \\ &= 2(AD + BC). \end{aligned}$$

Thus,

$$\frac{1}{2}M[1 - G(\epsilon)] = \frac{A}{B} = \frac{A}{B} + \frac{C}{D} - \frac{C}{D} = \frac{AD + BC}{BD} - \frac{C}{D} = \frac{M}{2} - \frac{C}{D}.$$

Therefore,

$$G(\epsilon) = \frac{2C}{MD} = \frac{2(M - m)}{M \left(1 + \epsilon + \sqrt{(1 - \epsilon)^2 + \frac{4m(n - m)}{M(n - M)}\epsilon} \right)}.$$

In this form, it can be easily shown that $G(\epsilon)$ is monotonically decreasing. To prove this, let

$$h(\epsilon) = 1 + \epsilon + \sqrt{(1 - \epsilon)^2 + 4\gamma\epsilon} \quad \text{where} \quad \gamma = \frac{m(n - m)}{M(n - M)}.$$

For $\epsilon \in (0, 1)$,

$$h'(\epsilon) = 1 + \frac{\epsilon - 1 + 2\gamma}{\sqrt{(1 - \epsilon)^2 + 4\gamma\epsilon}} > 0,$$

so, from (36), $G(\epsilon)$ is monotonically decreasing.

Now,

(i) For $m < M$, let

$$\beta = \frac{M - m}{M} \in (0, 1) \quad \text{and} \quad \gamma = \frac{m(n - m)}{M(n - M)} > 0.$$

Since $G(\epsilon)$ is monotonically decreasing for $\epsilon \in [0, 1]$ and $\epsilon_k \in (0, 1)$ for all $k \geq 1$, we have

$$0 < \frac{\beta}{1 + \sqrt{\gamma}} = G(1) < G(\epsilon_k) < G(0) = \beta < 1.$$

Hence, when k is sufficiently large,

$$0 < \frac{G(1)}{2} < \epsilon_{k+1} = G(\epsilon_k) + O(y_{k-1} - z_M) < \frac{G(0) + 1}{2} < 1.$$

Consequently, the convergence of $\{y_k\}_{k=1}^{\infty}$ to z_M is linear.

Moreover, (35) implies that

$$|\epsilon_{k+1} - G(\epsilon_k)| = |O(y_{k-1} - z_M)| \leq C(y_{k-1} - z_M) \quad \text{for } k \geq 1 \quad (37)$$

for some positive number C independent of k . Since $y_k \rightarrow z_M$, there exists an integer $N > 1$ such that

$$G(1) - C(y_{N-1} - z_M) > 0 \quad \text{and} \quad G(0) + C(y_{N-1} - z_M) < 1. \quad (38)$$

Let

$$c_1 = G(1) - C(y_{N-1} - z_M), \quad d_1 = G(0) + C(y_{N-1} - z_M),$$

$$c_{k+1} = G(d_k) - C(y_{N+k-1} - z_M), \quad d_{k+1} = G(c_k) + C(y_{N+k-1} - z_M),$$

for $k \geq 1$. Then,

- (i) $\{c_k\}$ is monotonically increasing and $\{d_k\}$ is monotonically decreasing, namely,

$$0 < c_1 < c_2 < c_3 < \cdots < c_k < \cdots < d_k < \cdots < d_3 < d_2 < d_1 < 1.$$

- (ii) $c_k < \epsilon_{N+k} < d_k$ for $k \geq 1$.

This can be shown by mathematical induction. For $k = 1$,

$$\begin{aligned} 0 &< c_1 = G(1) - C(y_{N-1} - z_M) && \text{(because of (38))} \\ &< G(\epsilon_N) - C(y_{N-1} - z_M) && (0 < \epsilon_N < 1 \text{ and } G \text{ is decreasing}) \\ &< \epsilon_{N+1} < G(\epsilon_N) + C(y_{N-1} - z_M) && \text{(because of (37))} \\ &< G(0) + C(y_{N-1} - z_M) = d_1 < 1. && \text{(because of (38))} \end{aligned}$$

So, both (i) and (ii) hold. Suppose the assertions hold for $k \geq 1$, that is,

$$0 < c_1 < \cdots < c_{k-1} < c_k < \epsilon_{N+k} < d_k < d_{k-1} < \cdots < d_1 < 1. \quad (39)$$

Then it follows from the monotonicity of both G and the sequence $\{y_k\}$ that

$$\begin{aligned} c_{k+1} &= G(d_k) - C(y_{N+k-1} - z_M) > G(d_{k-1}) - C(y_{N+k-2} - z_M) = c_k \\ d_{k+1} &= G(c_k) + C(y_{N+k-1} - z_M) < G(c_{k-1}) + C(y_{N+k-2} - z_M) = d_k. \end{aligned}$$

Furthermore,

$$\begin{aligned} c_{k+1} &= G(d_k) - C(y_{N+k-1} - z_M) \\ &< G(\epsilon_{N+k}) - C(y_{N+k-1} - z_M) && \text{(because of (39))} \\ &< \epsilon_{N+k+1} < G(\epsilon_{N+k}) + C(y_{N+k-1} - z_M) && \text{(because of (37))} \\ &< G(c_k) + C(y_{N+k-1} - z_M) = d_{k+1}. && \text{(because of (39))} \end{aligned}$$

Hence,

$$0 < c_1 < \cdots < c_k < c_{k+1} < \epsilon_{N+k+1} < d_{k+1} < d_k < \cdots < d_1 < 1.$$

We now let $c_k \rightarrow c$ and $d_k \rightarrow d$. Then $G(c) = d$ and $G(d) = c$. Since $G(\epsilon)$ is monotonically decreasing, $G^2 = G \circ G$ is monotonically increasing and can have at most one fixed point. It follows that $c = d$ and consequently $G(\omega) = \omega$, where

$$\omega = \lim_{k \rightarrow \infty} \epsilon_k = \lim_{k \rightarrow \infty} \frac{y_k - z_M}{y_{k-1} - z_M}.$$

Namely,

$$\omega = \frac{2(M - m)}{M \left(1 + \omega + \sqrt{(1 - \omega)^2 + \frac{4m(n-m)}{M(n-M)}\omega} \right)}$$

or, ω is a solution to the equation

$$g(x) \equiv \frac{n - M - m}{n - M}x^3 - x^2 - x + \frac{M - m}{M} = 0. \quad (40)$$

This equation has exactly one real solution in $(0, 1)$, since

$$\begin{aligned} g(0) &= \frac{M - m}{M} > 0, \\ g(1) &= \frac{n - M - m}{n - M} - 1 - 1 + \frac{M - m}{M} = \left[\frac{n - M - m}{n - M} - 1 \right] + \left[\frac{M - m}{M} - 1 \right] \\ &= \frac{-m}{n - M} + \frac{-m}{M} = \frac{-mn}{M(n - M)} < 0, \end{aligned}$$

and

$$g'(x) = 3\frac{n - M - m}{n - M}x^2 - 2x - 1 < 0 \quad \text{for } x \in (0, 1).$$

(ii) For $m = M$, from (36), $G(\epsilon_k) = 0$ for $k \geq 1$; accordingly, from (35),

$$\epsilon_{k+1} = O(y_{k-1} - z_M) \rightarrow 0, \quad \text{as } k \rightarrow \infty.$$

Thus the convergence of $\{y_k\}_{k=1}^{\infty}$ to z_M is super-linear. Further, from the derivation of y_{k+1} described in Chapter 1, y_{k-1} , y_k , and y_{k+1} satisfy the following two equations

$$\frac{M}{y_k - y_{k+1}} + \frac{n - M}{y_k - v} = q(y_k) = Q(y_k) + \frac{M}{y_k - z_M}$$

$$\frac{M}{y_{k-1} - y_{k+1}} + \frac{n - M}{y_{k-1} - v} = q(y_{k-1}) = Q(y_{k-1}) + \frac{M}{y_{k-1} - z_M}$$

for a certain number v . From the above two equations, we have

$$\frac{n - M}{y_k - v} = Q(y_k) - \frac{M(y_{k+1} - z_M)}{(y_k - z_M)(y_k - y_{k+1})} \quad (41)$$

$$\frac{n - M}{y_{k-1} - v} = Q(y_{k-1}) - \frac{M(y_{k+1} - z_M)}{(y_{k-1} - z_M)(y_{k-1} - y_{k+1})}. \quad (42)$$

Subtracting (42) from (41) and dividing the result by $(y_{k-1} - y_k)$ yields,

$$\frac{n - M}{(y_k - v)(y_{k-1} - v)} = R_1 - \frac{M(y_{k+1} - z_M)(y_{k-1} + y_k - y_{k+1} - z_M)}{(y_k - z_M)(y_k - y_{k+1})(y_{k-1} - z_M)(y_{k-1} - y_{k+1})}. \quad (43)$$

Substituting (41) and (42) into (43) and multiplying both sides by $(n - M)$, we have

$$\begin{aligned} & \left(Q(y_k) - \frac{M(y_{k+1} - z_M)}{(y_k - z_M)(y_k - y_{k+1})} \right) \left(Q(y_{k-1}) - \frac{M(y_{k+1} - z_M)}{(y_{k-1} - z_M)(y_{k-1} - y_{k+1})} \right) \\ &= (n - M)R_1 - \frac{M(n - M)(y_{k+1} - z_M)(y_{k-1} + y_k - y_{k+1} - z_M)}{(y_k - z_M)(y_k - y_{k+1})(y_{k-1} - z_M)(y_{k-1} - y_{k+1})}. \end{aligned}$$

Rewrite the above equation as follows:

$$\frac{y_{k+1} - z_M}{(y_k - z_M)(y_k - y_{k+1})(y_{k-1} - z_M)} \cdot E = (n - M)R_1 - Q(y_k)Q(y_{k-1}) \quad (44)$$

where

$$\begin{aligned} E &\equiv M(n - M) \frac{y_{k-1} + y_k - y_{k+1} - z_M}{y_{k-1} - y_{k+1}} - MQ(y_{k-1})(y_{k-1} - z_M) \\ &\quad - MQ(y_k)(y_k - z_M) \frac{y_k - y_{k+1}}{y_{k-1} - y_{k+1}} + M^2 \frac{y_{k+1} - z_M}{y_{k-1} - y_{k+1}} \\ &= M(n - M) \left(1 + \frac{\epsilon_k}{1 - \epsilon_k \epsilon_{k+1}} \right) - MQ(y_{k-1})(y_{k-1} - z_M) \\ &\quad - MQ(y_k)(y_k - z_M) \frac{\epsilon_k(1 - \epsilon_{k+1})}{1 - \epsilon_k \epsilon_{k+1}} + M^2 \frac{\epsilon_k \epsilon_{k+1}}{1 - \epsilon_k \epsilon_{k+1}} \\ &\rightarrow M(n - M) \quad \text{as } k \rightarrow \infty. \end{aligned}$$

It follows from (44) that

$$\begin{aligned}
 \frac{y_{k+1} - z_M}{(y_k - z_M)^2(y_{k-1} - z_M)} &= \left(\frac{y_k - y_{k+1}}{y_k - z_M} \right) \frac{(n-M)R_1 - Q(y_k)Q(y_{k-1})}{E} \\
 &= (1 - \epsilon_{k+1}) \frac{(n-M)R_1 - Q(y_k)Q(y_{k-1})}{E} \\
 &\rightarrow F \equiv \frac{(n-M)R^* - Q^2}{M(n-M)} < \infty \quad \text{as } k \rightarrow \infty.
 \end{aligned}$$

Here,

$$F = \frac{1}{M} \sum_{i=1}^{n-M} \frac{1}{(z_M - z^{(i)})^2} - \frac{1}{M(n-M)} \left(\sum_{i=1}^{n-M} \frac{1}{z_M - z^{(i)}} \right)^2 \geq 0.$$

If $F = 0$, then $z^{(i)} = z^{(1)}$, for $i = 1, \dots, n-M$. That is, $f(x)$ has only two distinct zeros: $z^{(1)}$ and z_M . Hence $x_{m-}^{(2)} = z_M$ and no convergence discussion is necessary. Consequently, only $f(x)$ which has more than two distinct zeros is considered. Then, $F > 0$. By the following lemma, the convergence rate of $\{y_k\}_{k=1}^{\infty}$ to z_M is indeed $\sqrt{2} + 1$. $\square$

Lemma 2.4 *Suppose a sequence $\{y_k\}_{k=1}^{\infty}$ converges to z . Let e_k be $|y_k - z|$. If*

$$e_{k+1} = M_k e_k^2 e_{k-1} \quad \text{and} \quad \lim_{k \rightarrow \infty} M_k = M^*$$

where $0 < M^* < \infty$, then the convergence rate of $\{y_k\}$ is $\sqrt{2} + 1$, i.e.,

$$\lim_{k \rightarrow \infty} \frac{e_{k+1}}{e_k^{\sqrt{2}+1}} = (M^*)^{1/\sqrt{2}}.$$

Proof. We assume first $M^* = 1$, that is, $\lim_{k \rightarrow \infty} M_k = 1$. Let

$$d_k = \ln e_{k+1} - (\sqrt{2} + 1) \ln e_k, \quad k \geq 1.$$

From $e_{k+1} = M_k e_k^2 e_{k-1}$,

$$\ln e_{k+1} = 2 \ln e_k + \ln e_{k-1} + \ln M_k.$$

So,

$$\begin{aligned}
 d_k &= 2 \ln e_k + \ln e_{k-1} + \ln M_k - (\sqrt{2} + 1) \ln e_k \\
 &= (1 - \sqrt{2}) \ln e_k + \ln e_{k-1} + \ln M_k \\
 &= (1 - \sqrt{2})[\ln e_k - (\sqrt{2} + 1) \ln e_{k-1}] + \ln M_k \\
 &= (1 - \sqrt{2})d_{k-1} + \ln M_k.
 \end{aligned}$$

Since $0 > 1 - \sqrt{2} > -1$ and $\ln M_k \rightarrow 0$ as $k \rightarrow \infty$, we have

$$d_k \rightarrow 0 \quad \text{as } k \rightarrow \infty.$$

That is,

$$\frac{e_{k+1}}{e_k^{\sqrt{2}+1}} \rightarrow 1 \quad \text{as } k \rightarrow \infty.$$

Now, if $M^* \neq 1$, let

$$e'_k = \sqrt{M^*} e_k \quad \text{and} \quad M'_k = \frac{M_k}{M^*}.$$

Then,

$$\begin{aligned}
 e'_{k+1} &= \sqrt{M^*} e_{k+1} = \frac{M_k}{M^*} (\sqrt{M^*} e_k)^2 (\sqrt{M^*} e_{k-1}) \\
 &= M'_k (e'_k)^2 e'_{k-1} \quad \text{and} \\
 \lim_{k \rightarrow \infty} M'_k &= 1.
 \end{aligned}$$

Therefore, it follows from the above derivation,

$$\lim_{k \rightarrow \infty} \frac{e'_{k+1}}{(e'_k)^{\sqrt{2}+1}} = 1$$

which implies

$$\lim_{k \rightarrow \infty} \frac{e_{k+1}}{e_k^{\sqrt{2}+1}} = \frac{(\sqrt{M^*})^{\sqrt{2}+1}}{\sqrt{M^*}} = (M^*)^{1/\sqrt{2}}. \quad \square$$

3 On symmetric tridiagonal eigenvalue problems

In this chapter, we shall use our quasi-Laguerre iteration derived in previous chapters to approximate all eigenvalues of symmetric tridiagonal matrices.

3.1 Evaluation of the logarithmic derivative f'/f of the determinant

Let T be a symmetric tridiagonal matrix of the form

$$T = [\beta_{i-1}, \alpha_i, \beta_i] = \begin{pmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & 0 \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \ddots \\ 0 & & & \beta_{n-2} & \alpha_{n-1} & \beta_{n-1} \\ & & & & \beta_{n-1} & \alpha_n \end{pmatrix}. \quad (45)$$

We may assume, without loss of generality, that T is unreduced; that is, $\beta_j \neq 0$, $j = 1, \dots, n-1$. For an unreduced T , the characteristic polynomial

$$f(\lambda) \equiv \det(T - \lambda I) \quad (46)$$

has only *real* and *simple* zeros ([14], p300). In order to use our quasi-Laguerre iteration developed in previous Chapters for finding zeros of $f(\lambda)$, or the eigenvalues of T , it is necessary to evaluate f and f' efficiently with satisfactory accuracy in the first place.

It is well known that the characteristic polynomial $f(\lambda) \equiv \det(T - \lambda I)$ of $T = [\beta_{i-1}, \alpha_i, \beta_i]$ and its derivative with respect to λ can be

evaluated by three-term recurrences ([14], p423):

$$\begin{cases} \rho_0 = 1, & \rho_1 = \alpha_1 - \lambda \\ \rho_i = (\alpha_i - \lambda)\rho_{i-1} - \beta_{i-1}^2 \rho_{i-2}, & i = 2, 3, \dots, n \end{cases} \quad (47)$$

$$\begin{cases} \rho'_0 = 0, & \rho'_1 = -1 \\ \rho'_i = (\alpha_i - \lambda)\rho'_{i-1} - \rho_{i-1} - \beta_{i-1}^2 \rho'_{i-2}, & i = 2, 3, \dots, n \end{cases} \quad (48)$$

and

$$f(\lambda) = \rho_n, \quad f'(\lambda) = \rho'_n.$$

However, these recurrences may suffer from a severe underflow/overflow problem and require constant testing and scaling. The following modified recurrence equations avoid the above problem subtly by computing the quotient $q(\lambda) = f'(\lambda)/f(\lambda)$ directly [8]. After all, only $q(\lambda)$ is really needed in the formulae (30), (31) and (32). Let

$$\xi_i = \frac{\rho_i}{\rho_{i-1}}, \quad \eta_i = -\frac{\rho'_i}{\rho_i},$$

$$\begin{cases} \xi_1 = \alpha_1 - \lambda, \\ \xi_i = \alpha_i - \lambda - \frac{\beta_{i-1}^2}{\xi_{i-1}}, & i = 2, 3, \dots, n, \end{cases} \quad (49)$$

$$\begin{cases} \eta_0 = 0, & \eta_1 = \frac{1}{\xi_1}, \\ \eta_i = \frac{1}{\xi_i} \left[(\alpha_i - \lambda)\eta_{i-1} + 1 - \left(\frac{\beta_{i-1}^2}{\xi_{i-1}} \right) \eta_{i-2} \right], & i = 2, 3, \dots, n \end{cases} \quad (50)$$

and

$$-\frac{f'(\lambda)}{f(\lambda)} = \eta_n.$$

To prevent the algorithm from breaking down when $\xi_i = 0$ for some $1 \leq i \leq n$, an extra check is provided:

- If $\xi_1 = 0$ (i.e., $\alpha_1 = \lambda$), set $\xi_1 = \beta_1^2 \varepsilon^2$;
- If $\xi_i = 0$, $i > 1$, set $\xi_i = \frac{\beta_{i-1}^2 \varepsilon^2}{\xi_{i-1}}$;

Algorithm DETEVL:

```

Input:   $T = [\beta_{i-1}, \alpha_i, \beta_i], \lambda$ 
Output:  $-\frac{f'(\lambda)}{f(\lambda)} = \eta_n$     (where  $f(\lambda) \equiv \det(T - \lambda I)$ )
           and  $\kappa(\lambda) = \text{neg\_count}$ 
           (=the number of eigenvalues of  $T$  less than  $\lambda$ )

Begin DETEVL
     $\xi_1 = \alpha_1 - \lambda$ 
    If  $\xi_1 = 0$  then  $\xi_1 = \beta_1^2 \varepsilon^2$ 
     $\eta_0 = 0, \quad \eta_1 = \frac{1}{\xi_1}, \quad \text{neg\_count} = 0$ 
    If  $\xi_1 < 0$  then  $\text{neg\_count} = 1$ 
    For  $i = 2 : n$ 
         $\xi_i = (\alpha_i - \lambda) - \left( \frac{\beta_{i-1}^2}{\xi_{i-1}} \right)$ 
        If  $\xi_i = 0$  then  $\xi_i = \left( \frac{\beta_{i-1}^2}{\xi_{i-1}} \right) \varepsilon^2$ 
        If  $\xi_i < 0$  then  $\text{neg\_count} = \text{neg\_count} + 1$ 
         $\eta_i = \frac{1}{\xi_i} \left[ (\alpha_i - \lambda) \eta_{i-1} + 1 - \left( \frac{\beta_{i-1}^2}{\xi_{i-1}} \right) \eta_{i-2} \right]$ 
    End for
End DETEVL

```

Figure 1: **Algorithm DETEVL**

where ε is the machine precision. A determinant evaluation subroutine DETEVL (see Figure 1) is formulated according to the recurrences (49) and (50). When $\xi_i, i = 1, \dots, n$, are known, the Sturm sequence is available ([9], p47). Thus, as a by-product, DETEVL also evaluates the number of eigenvalues of T which are less than λ . This number is denoted by $\kappa(\lambda)$.

Let $\lambda_1 < \lambda_2 < \dots < \lambda_n$ be the zeros of $f(\lambda)$ and $\hat{\lambda}_1 < \hat{\lambda}_2 < \dots < \hat{\lambda}_n$ be the zeros of the numerical approximation $\hat{f}(\lambda)$, it was shown in [8]

that

$$|fl(\hat{\lambda}_i) - \lambda_i| \leq \frac{5\varepsilon}{2} \max_j \{|\beta_j| + |\beta_{j+1}|\} + |\lambda_i|\varepsilon. \quad (51)$$

3.2 The split-merge process

Let

$$\lambda_1 < \lambda_2 < \cdots < \lambda_n$$

be the zeros of f in (46). Using the quasi-Laguerre iteration to approximate any λ_i , $i = 1, 2, \dots, n$, it is essential to provide a pair of starting points $x^{(0)}$ and $x^{(1)}$, being either $x^{(0)} < x^{(1)} < \lambda_i$ or $\lambda_i < x^{(1)} < x^{(0)}$, with no other λ_j 's lying between $x^{(0)}$, $x^{(1)}$ and λ_i . For this purpose, we *split* the matrix T into

$$\hat{T} = \begin{pmatrix} T_0 & 0 \\ 0 & T_1 \end{pmatrix} \quad (52)$$

where

$$T_0 = \begin{pmatrix} \alpha_1 & \beta_1 & & \\ \beta_1 & \ddots & \ddots & \\ & \ddots & \ddots & \beta_{k-1} \\ & & \beta_{k-1} & \alpha_k - \beta_k \end{pmatrix}, \quad T_1 = \begin{pmatrix} \alpha_{k+1} - \beta_k & \beta_{k+1} & & \\ \beta_{k+1} & \ddots & \ddots & \\ & \ddots & \ddots & \beta_{n-1} \\ & & \beta_{n-1} & \alpha_n \end{pmatrix}. \quad (53)$$

Obviously, the eigenvalues of $\hat{T}$ consist of eigenvalues of T_0 and T_1 . Without loss of generality, we may assume $\beta_i > 0$, for all $i = 1, 2, \dots, n-1$, since in (47)–(50), β_i 's always appear in their square form. The following interlacing property for this rank-one tearing is important to our algorithm.

Theorem 3.1 *Let $\lambda_1 < \lambda_2 < \cdots < \lambda_n$ and $\hat{\lambda}_1 \leq \hat{\lambda}_2 \leq \cdots \leq \hat{\lambda}_n$ be eigenvalues of T and $\hat{T}$ respectively. Then*

$$\hat{\lambda}_1 \leq \lambda_1 \leq \hat{\lambda}_2 \leq \lambda_2 \leq \cdots \leq \hat{\lambda}_n \leq \lambda_n < \hat{\lambda}_{n+1},$$

with the convention $\hat{\lambda}_{n+1} = \hat{\lambda}_n + 2\beta_k$.

Proof. See [4, Theorem 8.6.2, p462].

The eigenvalues of $\hat{T}$ will be used critically to approximate the eigenvalues of T by our quasi-Laguerre iteration. We shall call this procedure, splitting T into T_0 and T_1 of $\hat{T}$ and using eigenvalues of $\hat{T}$, consisting of eigenvalues of T_0 and T_1 , to approximate eigenvalues of T , the *split-merge* process, similar to Cuppen's divide-and-conquer strategy [2], of course.

To evaluate certain eigenvalue λ_i , $i = 1, 2, \dots, n$, of T , both $\hat{\lambda}_i$ and $\hat{\lambda}_{i+1}$ are suitable candidates for the first starting point of our algorithm, since by Theorem 3.1,

$$\hat{\lambda}_i \leq \lambda_i \leq \hat{\lambda}_{i+1}.$$

For the second starting point, let

$$c = \frac{\hat{\lambda}_i + \hat{\lambda}_{i+1}}{2}$$

and evaluate $\frac{f'(c)}{f(c)}$ by the subroutine DETEVL given in the last section. As we mentioned before, the Sturm sequence at c , which decides the position of c relative to λ_i , is a by-product of this evaluation. When $c > \lambda_i$, we may let $x_-^{(0)} = \hat{\lambda}_{i+1}$, $x_-^{(1)} = c$ and when $c < \lambda_i$, let $x_+^{(0)} = \hat{\lambda}_i$ and $x_+^{(1)} = c$. Then $\lambda_i < x_-^{(1)} < x_-^{(0)}$ or $x_+^{(0)} < x_+^{(1)} < \lambda_i$. Respectively, equations (31) and (32) with $m = 1$ are used to generate $x_{\pm}^{(k+1)}$ for $k \geq 1$, namely,

$$x_{\pm}^{(k+1)} = x_{\pm}^{(k)} + \frac{2 \left[n - (x_{\pm}^{(k-1)} - x_{\pm}^{(k)})q(x_{\pm}^{(k-1)}) \right]}{-(x_{\pm}^{(k-1)} - x_{\pm}^{(k)})R - 2q(x_{\pm}^{(k-1)}) \pm \sqrt{R \left[(x_{\pm}^{(k-1)} - x_{\pm}^{(k)})^2 R + 4(n-1) \right]}}.$$

And sequences $\{x_{\pm}^{(k)}\}_{k=1}^{\infty}$ are obtained which converge monotonically, either increasingly or decreasingly, to λ_i with ultimately convergence rate $\sqrt{2} + 1$ by Theorem 2.3.

The eigenvalues of $\hat{T}$ in (52) consist of eigenvalues of T_0 and T_1 in (53). To find eigenvalues of T_0 and T_1 , the split-merge process described above may be applied to T_0 and T_1 again. Indeed, the splitting process should be applied to T recursively (See Figure 2) until 2×2 and 1×1 matrices are reached.

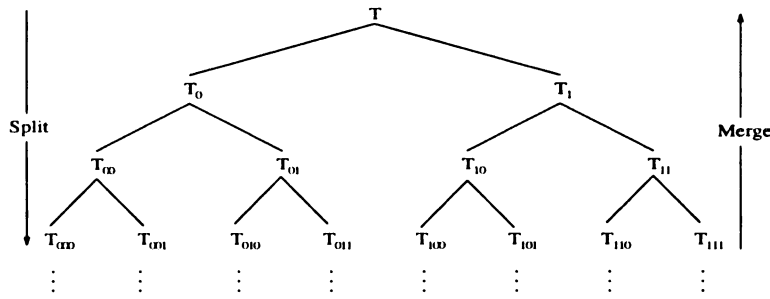


Figure 2: Split and merge processes

After T is well split into a tree structure as shown in Figure 2, the merging process in the reverse direction from 2×2 and 1×1 matrices can be started. More specifically, let T_{σ} be split into $T_{\sigma 0}$ and $T_{\sigma 1}$. Let $\hat{\lambda}_1^{\sigma}, \dots, \hat{\lambda}_m^{\sigma}$ be eigenvalues of $\hat{T}_{\sigma} = \begin{pmatrix} T_{\sigma 0} & 0 \\ 0 & T_{\sigma 1} \end{pmatrix}$ in ascending order. Then the quasi-Laguerre iteration is applied to the polynomial equation

$$f_{\sigma}(\lambda) \equiv \det[T_{\sigma} - \lambda I] = 0$$

from either $\hat{\lambda}_i^{(\sigma)}$ or $\hat{\lambda}_{i+1}^{(\sigma)}$ to obtain the corresponding eigenvalue λ_i^{σ} , $i = 1, 2, \dots, m$, by the merging process described above. This process is continued until T_0 and T_1 are merged into T . That is, in the final step all the eigenvalues of T are obtained by applying the quasi-Laguerre iteration to $f(\lambda) = \det(T - \lambda I)$ from eigenvalues of T_0 and T_1 .

3.3 Cluster and deflation

By Theorem 3.1, $\lambda_i \in (\hat{\lambda}_i, \hat{\lambda}_{i+1})$ for each $i = 1, \dots, n$ with the convention $\hat{\lambda}_{n+1} = \hat{\lambda}_n + 2\beta_k$. If $\hat{\lambda}_{i+1} - \hat{\lambda}_i$ is less than the error tolerance, then either $\hat{\lambda}_i$ or $\hat{\lambda}_{i+1}$ can be accepted as λ_i . In general, if $\hat{T}$ has a cluster of $r + 1$ very close eigenvalues, for instance, $\hat{\lambda}_{j+r} - \hat{\lambda}_j$ is less than the error tolerance for certain $1 \leq j \leq n - r$, then r eigenvalues $\lambda_j, \lambda_{j+1}, \dots, \lambda_{j+r-1}$ of T can be obtained free of computations. They can be set to any one of $\hat{\lambda}_j, \dots, \hat{\lambda}_{j+r}$.

Since the matrix T in (45) is unreduced, its eigenvalues are all simple. Therefore using $m = 1$ in the quasi-Laguerre iterations given in (31) or (32) seems appropriate in all cases to obtain ultimate super-linear convergence with convergence rate $\sqrt{2} + 1$. However, in some occasions, there may exist a group of $r > 1$ eigenvalues of T , say,

$$\lambda_{i+1} < \lambda_{i+2} < \dots < \lambda_{i+r}$$

which are relatively close to each other, comparing to the distance from the two starting points, say $x_+^{(0)} < x_+^{(1)} < \lambda_{i+1}$ as shown in Figure 3. Some of them may even be numerically indistinguishable.

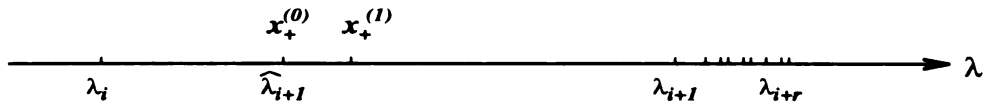


Figure 3: converging to relatively closer eigenvalues

Numerical evidence shows that to reach λ_{i+1} from $x_+^{(0)} (= \hat{\lambda}_{i+1})$ and $x_+^{(1)}$, it may take many steps of the quasi-Laguerre iteration with $m = 1$ before the super-linear convergence shows. In this situation, the quasi-Laguerre iteration with $m = r$ in (31) may be used to speed up the convergence. Exhibited in (29), the sequences $\{x_{m+}^{(k)}\}$ in (31) with

different m 's satisfy

$$x_+^{(k-1)} < x_{1+}^{(k)} < x_{2+}^{(k)} < \cdots < x_{r+}^{(k)} < \lambda_{i+r}.$$

Accordingly, $x_{r+}^{(k)}$ is relatively closer to λ_{i+1} . Therefore if the number r of those relatively closer zeros can be estimated, then instead of using $m = 1$ as usual, we may let $x_+^{(k)} = x_{r+}^{(k)}$. To estimate r in our algorithm, we put λ_j in the same group with λ_{i+1} if

$$|\hat{\lambda}_j - \hat{\lambda}_{i+2}| < 0.01|\hat{\lambda}_{i+2} - \hat{\lambda}_{i+1}|.$$

It might happen that $\lambda_{i+1} < x_+^{(k)}$, revealed by the information of the Sturm sequence given at $x_+^{(k)}$; in this case, the number r may be reduced. Therefore an overestimate of the number r causes essentially no harm because our algorithm can dynamically reduce r , while an underestimate may result in slow convergence.

3.4 Partial spectrum

From the strong interlacing property given in Theorem 3.1 one can easily obtain the following:

Proposition 3.2 *If $[a, b]$ contains k eigenvalues of $\hat{T}$, then $[a, b]$ contains at least $k-1$ and at most $k+1$ eigenvalues of T . More precisely, let $\kappa(x)$ be the number of eigenvalues of T which are less than $x \in \mathbb{R}$ and $\hat{\lambda}_{s+1}, \dots, \hat{\lambda}_{s+k}$ be all the eigenvalues of $\hat{T}$ in $[a, b]$. Then $s-1 \leq \kappa(a) \leq s$ and $s+k-1 \leq \kappa(b) \leq s+k$.*

In our algorithm, either $\hat{\lambda}_i$ or $\hat{\lambda}_{i+1}$ of $\hat{T}$ is used as the first starting point of our quasi-Laguerre iteration to obtain λ_i of T . To find eigenvalues of T in a given interval $[a, b]$, the eigenvalues of $\hat{T}$ in $[a, b]$ are known, say $\hat{\lambda}_{s+1}, \dots, \hat{\lambda}_{s+k}$. By evaluating $\kappa(a)$ and $\kappa(b)$, the actual member of

eigenvalues of T in $[a, b]$ is $\sigma = \kappa(b) - \kappa(a)$, so that $\lambda_{\kappa(a)+1}, \dots, \lambda_{\kappa(a)+\sigma}$ are the eigenvalues of T in $[a, b]$. By Proposition 3.2, $s - 1 \leq \kappa(a) \leq s$ and $s + k - 1 \leq \kappa(b) \leq s + k$, so, σ can either be $k - 1$, k , or $k + 1$. Thus, at most $k + 2$ values are needed to be considered as the first starting points to evaluate these σ eigenvalues of T . Let

$$\hat{\lambda}_s = a, \quad \hat{\lambda}_{s+1} = \hat{\lambda}_{s+1}, \quad \hat{\lambda}_{s+2} = \hat{\lambda}_{s+2}, \quad \dots, \quad \hat{\lambda}_{s+k} = \hat{\lambda}_{s+k}, \quad \hat{\lambda}_{s+k+1} = b.$$

Then σ values among them can serve as the first starting points which will lead to all σ eigenvalues of T in $[a, b]$.

In some occasions, the eigenvalues of T of interest are identified in magnitude, say the largest 20%, then they can be evaluated by using the largest 20% eigenvalues of $\hat{T}$ as starting points without computing the other 80% eigenvalues of both $\hat{T}$ and T .

3.5 Practical considerations in implementation

3.5.1 Stopping criteria

The following stopping criterion was suggested by Kahan [7] and used for our implementation:

$$|x^{(k+1)} - x^{(k)}|^2 \leq (|x^{(k)} - x^{(k-1)}| - |x^{(k+1)} - x^{(k)}|)\tau \quad (54)$$

where τ is the error tolerance. This criterion is based on the following observations. Let

$$q_k = \left| \frac{x^{(k+1)} - x^{(k)}}{x^{(k)} - x^{(k-1)}} \right|$$

then as $\{x^{(k)}\}_{k=1}^{\infty}$ converges to λ when $k \rightarrow \infty$, q_k is normally decreasing. Thus

$$|\lambda - x^{(k+1)}| = \left| \sum_{i=0}^{\infty} (x^{(k+2+i)} - x^{(k+1+i)}) \right| \leq \sum_{i=0}^{\infty} |x^{(k+2+i)} - x^{(k+1+i)}|$$

$$\begin{aligned}
&\leq |x^{(k+1)} - x^{(k)}| \sum_{i=1}^{\infty} q_k^i = \frac{q_k |x^{(k+1)} - x^{(k)}|}{1 - q_k} \\
&= \frac{|x^{(k+1)} - x^{(k)}|^2}{|x^{(k)} - x^{(k-1)}| - |x^{(k+1)} - x^{(k)}|}.
\end{aligned}$$

From (51), an obvious error tolerance at λ_i can be chosen as

$$\tau = \frac{5\varepsilon}{2} \max_j (|\beta_j| + |\beta_{j+1}|) + |x^{(k+1)}| \varepsilon.$$

3.5.2 Initial points for the quasi-Laguerre iteration

At every stage of the merging process, we evaluate the eigenvalues $\lambda_1 < \dots < \lambda_m$ of an $m \times m$ submatrix, given m initial values $\hat{\lambda}_1 \leq \dots \leq \hat{\lambda}_m$ and an upper bound that interlace those m eigenvalues:

$$\hat{\lambda}_1 \leq \lambda_1 \leq \hat{\lambda}_2 \leq \lambda_2 \leq \dots \leq \hat{\lambda}_m \leq \lambda_m \leq \hat{\lambda}_{m+1}$$

(see Theorem 3.1). To evaluate an eigenvalue λ_i by the quasi-Laguerre iteration, two initial points are needed on the same side of λ_i without any other eigenvalues between them. However, for $i = 1, \dots, m$, the interlacing property

$$\hat{\lambda}_i \leq \lambda_i \leq \hat{\lambda}_{i+1}$$

provides only one initial point from each side. The simplest way of obtaining the other initial point, preferably in the interval $(\hat{\lambda}_i, \hat{\lambda}_{i+1})$, is to bisect the interval by choosing $c = \frac{\hat{\lambda}_i + \hat{\lambda}_{i+1}}{2}$ and evaluate $\frac{f'(c)}{f(c)}$ along with the Sturm sequence at c by the algorithm DETEVL as described in Section 3.2. If $c < \lambda_i$ then $\hat{\lambda}_i$ and c are a suitable pair of initial points for approximating λ_i . Otherwise, the pair c and $\hat{\lambda}_{i+1}$ will be used.

While the above approach is the simplest, it may not be the most efficient. Several improvements are made in our practical implementation. First, since f'/f is evaluated at every initial value $\hat{\lambda}_j$ ($j = 1, \dots, m$), one step of Newton's iteration can be performed without extra cost and

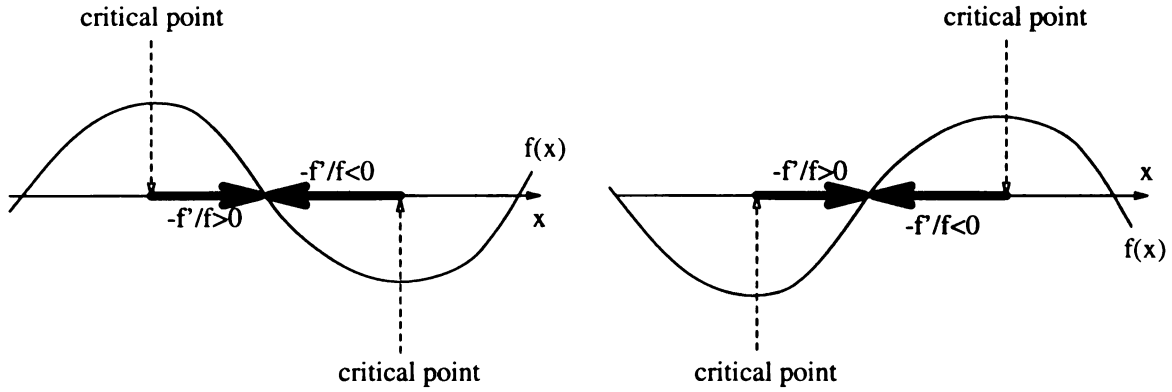


Figure 4: The two initial points should be between one of the nearby critical point of f and the zero. This can be identified by the sign pattern of $-f'/f$.

the result can be considered as a candidate, besides c , for $x^{(1)}$. Secondly and more importantly, from our computing experience, the high order convergence of the quasi-Laguerre iteration occurs only when there is no critical point of f (i.e. zero of f') between $x^{(0)}$ and λ_i (see Figure 4). In other words, if $x^{(0)}$ is to the left (resp. right) of λ_i , then it is desirable that $-f'(x^{(0)})/f(x^{(0)})$ is positive (resp. negative). If there is one or more critical point in $[\hat{\lambda}_i, \hat{\lambda}_{i+1}]$, then bisection or one step Newton's iteration is used repeatedly until the above requirement at $x^{(0)}$ is satisfied. Based on these observations, our procedure for determining a pair of initial points among different choices to achieve the best possible efficiency in practice is summarized below and the algorithm INIPTS is given in Figure 5. For notational convenience, we assume the eigenvalue λ_i is in the interval $[a, b]$ and there are no other eigenvalues in this interval.

START.

1. If $-\frac{f'(a)}{f(a)} < 0$ and $-\frac{f'(b)}{f(b)} > 0$, then neither a nor b should be used as an initial point (there are critical points of f in both (a, λ_i) and (λ_i, b)). Set $c = \frac{a+b}{2}$. Evaluate $-\frac{f'(c)}{f(c)}$, the Sturm

sequence at c is available in the process. Reset $[a, b]$ to be $[c, b]$ or $[a, c]$ in accordance with $c < \lambda_i$ or $c > \lambda_i$ respectively. Go to START.

2. If $-\frac{f'(a)}{f(a)} < 0$ (there is a critical point between a and λ_i , so a should not be used as an initial point) and $-\frac{f'(b)}{f(b)} < 0$, then b can be one of the initial point. Set $c = \max \left\{ b - \frac{f'(b)}{f(b)}, \frac{a+b}{2} \right\}$ and evaluate $-\frac{f'(c)}{f(c)}$. If $c \in (\lambda_i, b)$, then b and c will be used as a pair of initial points; go to END. Otherwise, reset $[a, b]$ to be $[c, b]$; go to START.
3. If $-\frac{f'(a)}{f(a)} > 0$ and $-\frac{f'(b)}{f(b)} > 0$, then a can be an initial point but b can not (there is a critical point of f between b and λ_i). Set $c = \min \left\{ a - \frac{f'(a)}{f(a)}, \frac{a+b}{2} \right\}$ and evaluate $-\frac{f'(c)}{f(c)}$. If $c \in (a, \lambda_i)$, then a and c are a good pair of initial points; go to END. Otherwise, reset $[a, b]$ to be $[a, c]$; go to START.
4. If $-\frac{f'(a)}{f(a)} > 0$ and $-\frac{f'(b)}{f(b)} < 0$, then there is no critical point of f in $[a, b]$. Thus either a or b can be used as one of the initial points. Set

$$c = \begin{cases} \min \left\{ a - \frac{f'(a)}{f(a)}, \frac{a+b}{2} \right\}, & \text{if } \left| \frac{f'(a)}{f(a)} \right| \leq \left| \frac{f'(b)}{f(b)} \right| \\ \max \left\{ b - \frac{f'(b)}{f(b)}, \frac{a+b}{2} \right\}, & \text{otherwise.} \end{cases}$$

and evaluate $-\frac{f'(c)}{f(c)}$. Choose $\{a, c\}$ or $\{c, b\}$ to be the initial points depending on $c < \lambda_i$ or $c > \lambda_i$ respectively; go to END.

END.

Algorithm INIPTS

Input: subscript i , initial end points $\hat{\lambda}_i, \hat{\lambda}_{i+1}$.

Output: starting iterates $x^{(0)}, x^{(1)}$ and $\kappa(x^{(0)}), \kappa(x^{(1)}), -\frac{f'(x^{(0)})}{f(x^{(0)})}, -\frac{f'(x^{(1)})}{f(x^{(1)})}$.

Begin INIPTS

$(a, b) = (\hat{\lambda}_i, \hat{\lambda}_{i+1})$; Evaluate $-\frac{f'(a)}{f(a)}, \kappa(a), -\frac{f'(b)}{f(b)}, \kappa(b)$ by DETEVL;

(##) If $-\frac{f'(a)}{f(a)} < 0$ and $-\frac{f'(b)}{f(b)} > 0$ then

$c = \frac{a+b}{2}$; Evaluate $-\frac{f'(c)}{f(c)}, \kappa(c)$ by DETEVL;

$(a, b) = \begin{cases} (c, b), & \text{if } c < \lambda_i \text{ (i.e. } \kappa(c) < i) \\ (a, c), & \text{if } c > \lambda_i \text{ (i.e. } \kappa(c) \geq i) \end{cases}$; Goto (##);

else if $-\frac{f'(a)}{f(a)} < 0$ and $-\frac{f'(b)}{f(b)} < 0$ then

$c = \max\left\{b - \frac{f'(b)}{f(b)}, \frac{a+b}{2}\right\}$; Evaluate $-\frac{f'(c)}{f(c)}, \kappa(c)$ by DETEVL;

If $c > \lambda_i$ then $(x^{(0)}, x^{(1)}) = (b, c)$; Goto (#) ;

else $(a, b) = (c, b)$; Goto (##) ;

else if $-\frac{f'(a)}{f(a)} > 0$ and $-\frac{f'(b)}{f(b)} > 0$ then

$c = \min\left\{a - \frac{f'(a)}{f(a)}, \frac{a+b}{2}\right\}$; Evaluate $-\frac{f'(c)}{f(c)}, \kappa(c)$ by DETEVL;

If $(c < \lambda_i)$ then $(x^{(0)}, x^{(1)}) = (a, c)$; Goto (#) ;

else $(a, b) = (a, c)$; Goto (##) ;

else

$c = \begin{cases} \min\left\{a - \frac{f'(a)}{f(a)}, \frac{a+b}{2}\right\}, & \text{if } \left|\frac{f'(a)}{f(a)}\right| \leq \left|\frac{f'(b)}{f(b)}\right| \\ \max\left\{b - \frac{f'(b)}{f(b)}, \frac{a+b}{2}\right\}, & \text{otherwise} \end{cases}$

Evaluate $-\frac{f'(c)}{f(c)}, \kappa(c)$ by DETEVL;

set $(x^{(0)}, x^{(1)}) = \begin{cases} (a, c), & \text{if } c < \lambda_i \\ (b, c), & \text{if } c > \lambda_i \end{cases}$

End if ;

(#) End INIPTS

Figure 5: **Algorithm INIPTS**

3.5.3 An alternative approach for clusters of eigenvalues

The multiplicity estimation and the quasi-Laguerre iterations (31) and (32) with $m > 1$ can successfully overcome the slow convergence when a cluster of eigenvalues occur. An alternative approach without multiplicity estimation as well as dynamical adjustments of the multiplicities can also work very efficiently in practice:

By Theorem 2.3, when the quasi-Laguerre iteration for simple zero ($m = 1$) is used to approximate an M -fold zero x^* ($M \geq 2$), or a cluster of M simple zeros, of a polynomial f of degree n , it converges linearly with a ratio $q_{n,M}$ which is the only solution of

$$\frac{n - M - 1}{n - M}x^3 - x^2 - x + \frac{M - 1}{M} = 0$$

in $(0, 1)$. The following theorem gives the properties of $q_{n,M}$ for $M \geq 2$ and $n \geq M + 1$.

Theorem 3.3 (i) For M fixed, $q_{n,M}$ increases as n increases ($n > M$).

(ii) For a given n , $q_{n,M} \geq q_{n,2}$ if $2 \leq M \leq n - 2$.

(iii) When $M = n - 1$, $q_{n,n-1}$ increases as n increases ($n \geq 3$).

Proof. (i) For a fixed M , write

$$g_n(x) \equiv \frac{n - M - 1}{n - M}x^3 - x^2 - x + \frac{M - 1}{M}.$$

Then when $n_1 > n_2$,

$$g_{n_1}(x) - g_{n_2}(x) = \left(\frac{1}{n_2 - M} - \frac{1}{n_1 - M} \right) x^3 > 0 \text{ for } x \in (0, 1).$$

So, $q_{n_1,M} > q_{n_2,M}$.

(ii) For a given n , write

$$g_M(x) \equiv \frac{n - M - 1}{n - M}x^3 - x^2 - x + \frac{M - 1}{M}.$$

When $2 \leq M \leq \frac{n}{2}$,

$$\begin{aligned} g_M(x) - g_2(x) &= \left(\frac{1}{n-2} - \frac{1}{n-M} \right) x^3 + \frac{1}{2} - \frac{1}{M} \\ &\geq \frac{1}{2} + \frac{1}{n-2} - \frac{1}{M} - \frac{1}{n-M} \\ &= \frac{n}{2(n-2)} - \frac{n}{M(n-M)} \geq 0 \quad \text{for } x \in (0, 1). \end{aligned}$$

So, $q_{n,M} \geq q_{n,2}$.

If $\frac{n}{2} < M \leq n-2$, then $2 \leq n-M < \frac{n}{2}$. It follows from above, $q_{n,n-M} \geq q_{n,2}$. On the other hand,

$$g_M(x) - g_{n-M}(x) = \left(\frac{1}{n-M} - \frac{1}{M} \right) (1-x^3) > 0 \quad \text{for } x \in (0, 1).$$

Then $q_{n,M} > q_{n,n-M}$. Hence, $q_{n,M} > q_{n,n-M} \geq q_{n,2}$.

(iii) If $M = n-1$, write

$$g_n(x) \equiv \frac{n-M-1}{n-M} x^3 - x^2 - x + \frac{M-1}{M} = -x^2 - x + 1 - \frac{1}{n-1}.$$

Then when $n_1 > n_2$,

$$g_{n_1}(x) - g_{n_2}(x) = \frac{1}{n_2-1} - \frac{1}{n_1-1} > 0 \quad \text{for } x \in (0, 1).$$

So, $q_{n_1, n_1-1} > q_{n_2, n_2-1}$. $\square$

For $n \geq 3$ and $M \geq 2$, by Theorem 3.3 (iii),

$$q_{n,M} \geq q_{3,2} \quad \text{when } M = n-1,$$

and by Theorem 3.3 (i), (ii),

$$q_{n,M} \geq q_{n,2} \geq q_{3,2} \quad \text{when } M \leq n-2.$$

In any case, $q_* \equiv q_{3,2}$, the zero of $x^2 + x - 0.5 = 0$ in $(0, 1)$, is the smallest linear ratio for multiple zeros. Let $x^{(1)}, x^{(2)}, \dots, x^{(k)}, \dots$ be

the monotone sequence of the quasi-Laguerre iterates. When $\{x^{(k)}\}$ converges super-linearly, then

$$q_k \equiv \frac{|x^{(k+1)} - x^{(k)}|}{|x^{(k)} - x^{(k-1)}|} \longrightarrow 0 \quad \text{as } k \rightarrow \infty,$$

so, q_k should be smaller than q_* when k is sufficiently large. Therefore, when $q_k > q_*$, linear convergence seems apparent. In this case, for sufficiently large k ,

$$\frac{|x^{(k+1)} - x^*|}{|x^{(k)} - x^*|} \approx q_{n,M} \quad \text{or} \quad |x^{(k+1)} - x^*| \approx q|x^{(k)} - x^*| \quad \text{where } q \equiv q_{n,M},$$

and

$$q_k = \frac{|x^{(k+1)} - x^{(k)}|}{|x^{(k)} - x^{(k-1)}|} = \frac{|x^{(k)} - x^*| - |x^{(k+1)} - x^*|}{|x^{(k-1)} - x^*| - |x^{(k)} - x^*|} \approx \frac{(1-q)|x^{(k)} - x^*|}{(1-q)|x^{(k-1)} - x^*|} \approx q.$$

In practice, q_k 's can be very close to q after several iterations. If $q_j = q$ for $j \geq k$ then

$$|x^{(j+1)} - x^*| = q|x^{(j)} - x^*|, \quad \text{for } j \geq k$$

and

$$x^* = x^{(k)} + \frac{1}{1-q}(x^{(k+1)} - x^{(k)}).$$

So, at the k -th iterate $x^{(k)}$, if linear convergence is revealed, namely $q_k > q_*$, then instead of performing the regular quasi-Laguerre iteration

$$x^{(k+1)} = x^{(k)} + \delta_k$$

where δ_k is the correction $x^{(k+1)} - x^{(k)}$ from $x^{(k)}$ calculated by the quasi-Laguerre iteration with $m = 1$, one can accelerate the iteration by setting

$$x^{(k+1)} = x^{(k)} + \frac{\delta_k}{1 - q_k}.$$

The accelerated iteration can be summarized as follows: Assume we are evaluating the eigenvalue $\lambda^* \in [a, b]$ and initial points $x^{(0)}, x^{(1)}$ have

been obtained by the process described in Section 3.5.2 along with data $-\frac{f'(x^{(0)})}{f(x^{(0)})}$ and $-\frac{f'(x^{(1)})}{f(x^{(1)})}$.

- (#) For $k = 1, 2, \dots$ do (while the stopping criteria is not satisfied),
1. Evaluate $\delta_k = x^{(k+1)} - x^{(k)}$ using the quasi-Laguerre formula with $m = 1$.
 2. If $q \stackrel{\text{def}}{=} \frac{\delta_k}{\delta_{k-1}} < q_*$, then most likely the target eigenvalue λ^* is neither a multiple zero nor a member of any cluster. Go to # and continue the loop.
- Otherwise,
- (a) Let $c = x^{(k)} + \frac{\delta_k}{1-q}$; and check for a possible overshoot by evaluating the Sturm sequence. If c overshoots, then reset $c = x^{(k)} + \frac{\delta_k(1-q^l)}{1-q}$ for $l = 8, 4, 2, 1$ until overshooting disappears.
 - (b) set $x^{(k+1)} = c$ and go to #.

End for

The algorithm ACCQ-LAG in Figure 6 summarizes the fundamental features of the acceleration process described in this section. Table 1 gives the comparison between the original quasi-Laguerre iteration and the accelerated quasi-Laguerre iteration. The Wilkinson matrix W_{99}^+ is used here with the starting points $x^{(0)} = 11.5$ and $x^{(1)} = L_-(x^{(0)}) = 11.27$ targeting $\lambda_{23} = 11.0 + 5 \times 10^{-15}$ of W_{99}^+ , while $\lambda_{22} = 11.0 - 5 \times 10^{-15}$.

3.6 Numerical tests

Our algorithm is implemented and tested on SPARC stations with IEEE floating point standard. The machine precision is $\varepsilon \approx 2.2 \times 10^{-16}$.

Algorithm AccQ-LAG

Input: subscript i , starting iterates $x^{(0)}, x^{(1)}$ with $\kappa(x^{(0)}), \kappa(x^{(1)}), -\frac{f'(x^{(0)})}{f(x^{(0)})}$ and $-\frac{f'(x^{(1)})}{f(x^{(1)})}$ evaluated by INIPTS.

output: the i -th eigenvalue λ_i of T .

Define $q_* = q_{3,2}$: the positive solution of $x^2 + x - 0.5 = 0$
(the smallest linear ratio for multiple zeros);

Begin AccQ-LAG

For $k = 1, 2, \dots$

$c = \begin{cases} u_+, & \text{if } x^{(k)} < \lambda_i \\ u_-, & \text{if } x^{(k)} > \lambda_i \end{cases}$ ($u_{\pm}$ are from quasi-Lag.formulae);

$\delta_k = c - x^{(k)}$; $q = \frac{\delta_k}{\delta_{k-1}}$; $tol = 2.5\varepsilon [\max_j (|\beta_j| + |\beta_{j+1}|)] + |x^{(k)}|\varepsilon$;

If $(|\delta_k| < tol)$ or $(k > 2 \text{ and } \frac{\delta_k^2}{|\delta_{k-1}||-\delta_k|} < tol)$ Goto (#) ;

If $(q > q_*)$ then

$c = x^{(k)} + \frac{\delta_k}{1-q}$; evaluate $-\frac{f'(c)}{f(c)}$ and $\kappa(c)$ by DETEVL;

If $(c \text{ is on the same side of } \lambda_i \text{ as } x^{(k)}, x^{(k-1)})$ then

Goto (##);

else, For $l = 8, 4, 2, 1$,

$c = x^{(k)} + \frac{\delta_k(1-q^l)}{1-q}$;

evaluate the Sturm sequence $\kappa(c)$;

If $(c \text{ is on the same side of } \lambda_i \text{ as } x^{(k)}, x^{(k-1)})$

Goto (##);

End for

End if

(##) $x^{(k+1)} = c$;

End for

(#) End AccQ-LAG

Figure 6: Algorithm AccQ-LAG

k	$x^{(k)}$ (original)	$q_k = \frac{ x^{(k+1)} - x^{(k)} }{ x^{(k)} - x^{(k-1)} }$	$x^{(k)}$ (accelerated)
1	11 .500000000000000		11. 500000000000000
2	11 .270700712327294		11. 270700712327294
3	11 .149428542073966	0. 706618	11.0 13287361699255
4	11.0 63735435125402	0.4 37604	11.000 124299930121
5	11.0 26235830973979	0.4 17533	11.0000 61856842477
6	11.0 10578521008206	0.40 3427	11.000000 198368395
7	11.00 4261938886133	0.40 3112	11.0000000 99023529
8	11.00 1715645778917	0.4025 55	11.0000000000 12077
9	11.000 690622605815	0.4025 56	11.00000000000 6040
10	11.000 277993423158	0.40252 7	11.00000000000 2304
11	11.000 111899071095	0.40252 5	11.000000000000 156
12	11.0000 45041905516	0.40252 3	11.000000000000 64
13	11.0000 18130368451	0.402522	
14	11.00000 7297872838	0.402522	
15	11.00000 2937554150	0.402522	
16	11.00000 1182429996	0.402522	
17	11.000000 475954003	0.402522	
18	11.000000 191581922	0.402522	
19	11.0000000 77115924	0.402522	
20	11.0000000 31040850	0.402522	
21	11.0000000 12494622	0.402522	
22	11.00000000 5029360	0.402522	
23	11.00000000 2024429	0.402522	
24	11.000000000 814877	0.402522	
25	11.000000000 328006	0.402522	
26	11.000000000 132029	0.402520	
27	11.0000000000 53145	0.40252 1	
28	11.0000000000 21393	0.40252 6	
29	11.00000000000 8612	0.40252 1	
30	11.00000000000 3467	0.402 477	
31	11.00000000000 1396	0.40 1680	
32	11.00000000000 565	0. 399604	
33	11.00000000000 233	0. 381937	
34	11.00000000000 107	0. 305425	

Table 1: Acceleration of the quasi-Laguerre iteration for W_{99}^+ . For each number in the table, the digits before a space are correct.

3.6.1 Testing matrices

There are 12 types of matrices employed in testing our algorithm. In the following description of these matrix types, α_i , $i = 1, \dots, n$, denote the diagonal entries and β_i , $i = 1, \dots, n - 1$, are the sub-diagonal entries.

Matrices with known eigenvalues

1. Toeplitz matrices $[b, a, b]$. Exact eigenvalues:
 $\{a + 2b \cos \frac{k\pi}{n+1}\}_{1 \leq k \leq n}$ ([5], Example 7.4, p137).
2. $\alpha_1 = a - b$, $\alpha_i = a$ for $i = 2, \dots, n - 1$, $\alpha_n = a + b$. $\beta_j = b$, $j = 1, \dots, n - 1$. Exact eigenvalues:
 $\{a + 2b \cos \frac{(2k-1)\pi}{2n}\}_{1 \leq k \leq n}$ ([5], Example 7.6, p138).
3. $\alpha_i = \begin{cases} a & \text{for odd } i \\ b & \text{for even } i \end{cases}$, $\beta_i = 1$. Exact eigenvalues :
 $\left\{ \frac{a+b \pm \sqrt{(a-b)^2 + 16 \cos^2 \frac{k\pi}{n+1}}}{2} \right\}_{1 \leq k \leq n/2}$ (plus $\{a\}$ if n is odd)
([5], Example 7.8 and 7.9, p139).
4. $\alpha_i = 0$, $\beta_i = \sqrt{i(n-i)}$. Exact eigenvalues:
 $\{-n + 2k - 1\}_{1 \leq k \leq n}$ ([5], Example 7.10, p140).
5. $\alpha_i = -[(2i-1)(n-1) - 2(i-1)^2]$, $\beta_i = i(n-i-2)$. Exact eigenvalues: $\{-k(k-1)\}_{1 \leq k \leq n}$ ([5], Example 7.11, p141).

Wilkinson and random matrices

6. Wilkinson matrices W_n^+ . $\beta_i = 1$,
 $\alpha_i = \begin{cases} n/2 - i + 1 & \text{for even } n \text{ and } 1 \leq i \leq n/2 \\ i - n/2 & \text{for even } n \text{ and } n/2 < i \leq n \\ (n-1)/2 - i + 1 & \text{for odd } n \text{ and } 1 \leq i \leq (n+1)/2 \\ i - (n+1)/2 & \text{for odd } n \text{ and } (n+1)/2 < i \leq n \end{cases}$

([14], pp308-309). Most of the eigenvalues are in pairs, consisting of two numerically indistinguishable eigenvalues.

7. Random matrices. α_i 's and β_i 's are random numbers in $[0, 1]$.

LAPACK testing matrices (generated by the LAPACK test matrix generator [1])

8. Matrices with eigenvalues evenly distributed between its smallest and largest eigenvalues.
9. Matrices with geometrically distributed eigenvalues. Namely, eigenvalues can be written as $\{q^k\}_{1 \leq k \leq n}$ for some $q \in (0, 1)$.
10. Matrices with an eigenvalue 1 and the remaining eigenvalues in $(-\varepsilon, \varepsilon)$.
11. Matrices with eigenvalues evenly distributed in the interval $(0, 1]$ except one eigenvalue with very small magnitude.
12. Matrices with an eigenvalue 1 and the rest of the eigenvalues are evenly distributed in a small interval $[10^{-12} - \varepsilon, 10^{-12} + \varepsilon]$.

3.6.2 Speed test in evaluating eigenvalues without computing eigenvectors

We compare the performance of the following codes for evaluating eigenvalues of an $n \times n$ matrix:

- (1) Q-LAG: our split-merge algorithm using the quasi-Laguerre iteration. Storage requirement: $9n$;
- (2) LAG: split-merge algorithm using Laguerre's iteration developed by T.Y. Li and Z. Zeng [8]. Storage requirement: $9n$;
- (3) B/M: bisection/multisection subroutine DSTEBSZ in LAPACK. Storage requirement: $12n$;

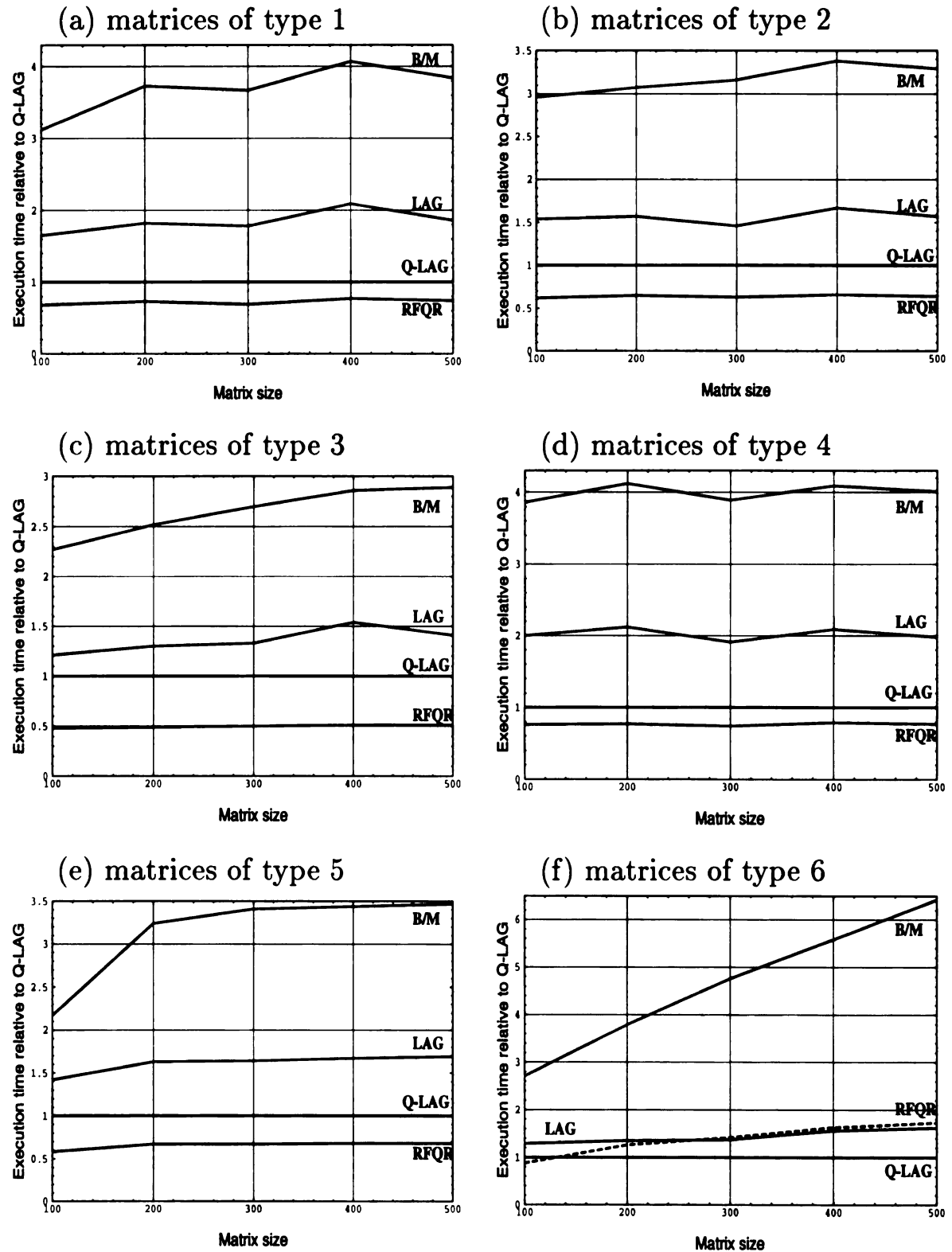


Figure 7: Relative execution time in evaluating all eigenvalues without computing eigenvectors. B/M: DSTEBZ; LAG: Laguerre's Iteration; Q-LAG: the quasi-Laguerre Iteration; RFQR: DSTERF.

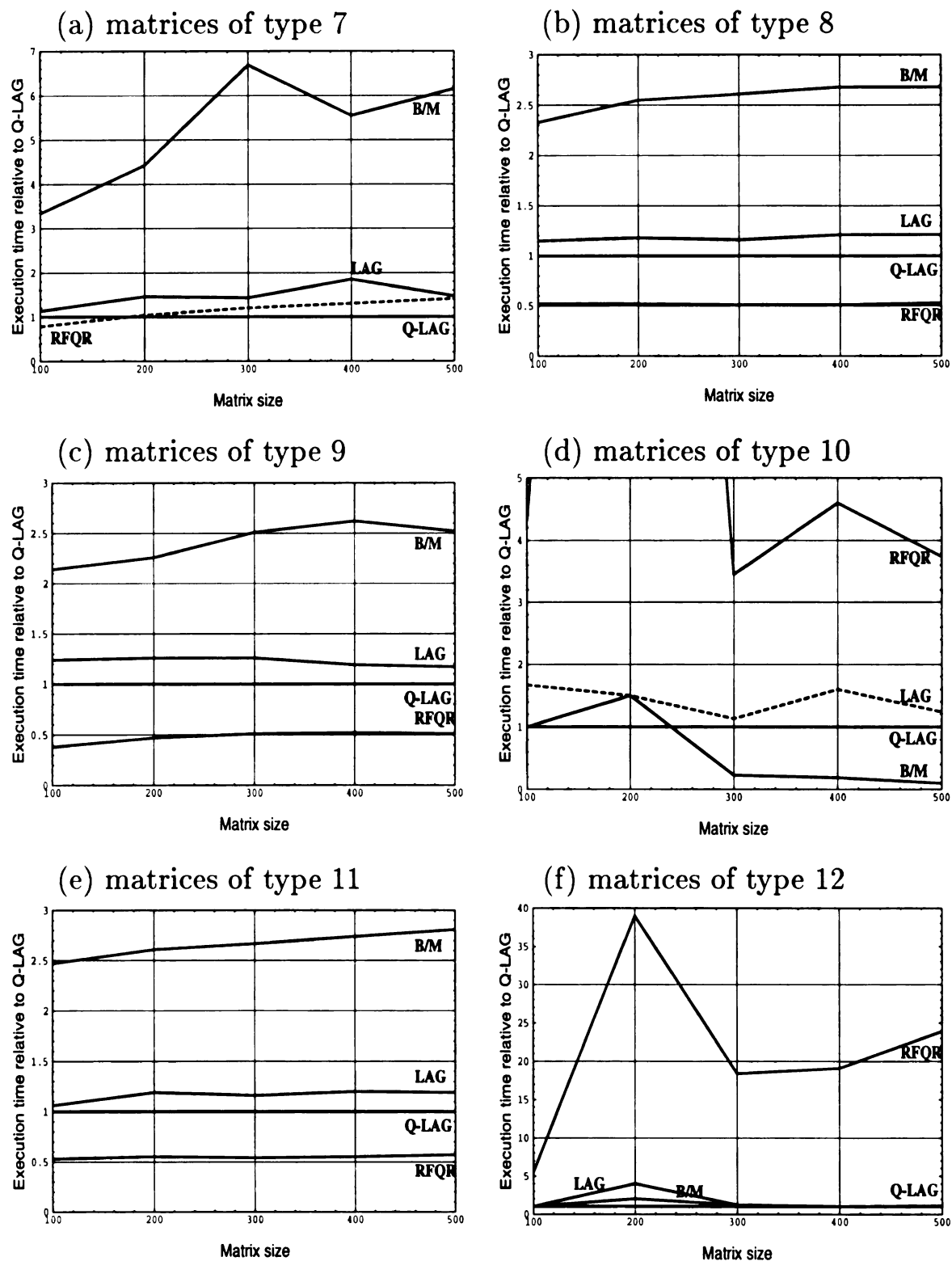


Figure 8: Relative execution time in evaluating all eigenvalues without computing eigenvectors. B/M: DSTEBZ; LAG: Laguerre's Iteration; Q-LAG: the quasi-Laguerre Iteration; RFQR: DSTERF.

- (4) RFQR: root-free QR routine DSTERF in LAPACK, as recommended in LAPACK for evaluating eigenvalues only. Storage requirement: $2n$.

The result of this test for matrices of type 1 through type 6 is given in Figure 7 and the result for matrices of types 7 through types 12 in Figure 8. Matrices of type 9 to type 12 involve tiny eigenvalues or clusters and are used more in stability tests (Please see Section 3.6.4). The speed comparison in this category may not be relevant. In particular, matrices of types 10 and 12 have big and dense clusters, and intensive deflations make Q-LAG as well as LAG outscore RFQR in speed by a wide margin.

For all other matrices (type 1 through type 8), Q-LAG is much faster than LAG (especially for types 1, 4 and 5) and LAG is much faster than B/M. RFQR is apparently the fastest code in almost all cases, except for matrices of type 6 and type 7 where Q-LAG is about 70% and 40% ($n = 500$) faster respectively. Overall, Q-LAG performs very similarly to RFQR and is much more competitive than B/M. Taking into account Q-LAG's flexibility in evaluating partial spectrum and the potential in parallel computing, Q-LAG is by far the most efficient algorithm among all the algorithms being tested here.

3.6.3 Speed test in evaluating eigenpairs

We compare the performance of the following codes for evaluating the eigenpairs of an $n \times n$ matrix:

- (1) Q-LAG: our split-merge algorithm using the quasi-Laguerre iteration plus the inverse iteration subroutine DSTEIN in LAPACK. Storage requirement: $n^2 + O(n)$;

- (2) B/M: DSTEBZ plus DSTEIN. Storage requirement: $n^2 + O(n)$;
- (3) INV: DSTEIN with eigenvalues computed by Q-LAG (for analysis purposes);
- (4) D&C: divide and conquer subroutine DSTEDC in LAPACK. Storage requirement: $3n^2 + O(n)$;
- (5) QR: QR routine DSTEQR in LAPACK, as recommended in LAPACK for evaluating eigenpairs only. Storage requirement: $n^2 + O(n)$.

The storage requirement for D&C is relatively higher than those which use the inverse iteration code. The results of evaluations of all eigenpairs and 1/3 of largest eigenvalues with corresponding eigenvectors are listed in columns *A* and *B* respectively in Table 2. In comparing the execution time in evaluating all eigenpairs, our algorithm leads QR substantially. For matrices of types 6, 9 and 10 where a number of clusters exist, our algorithm is slower than D&C due to drastic deflations of clusters. But this sort of matrices scarcely occurs in practice, they are often used for testing the robustness of an algorithm. For all other types of matrices, our algorithm leads D&C for large matrices ($n \geq 300$), while for small matrices, D&C is faster since it takes much longer time for INV to compute eigenvectors in our algorithm.

Only Q-LAG, LAG and B/M can be used efficiently to evaluate 1/3 of the largest eigenvalues and their corresponding eigenvectors, where D&C and QR must evaluate all the eigenpairs to obtain the largest 1/3. This makes our algorithm more flexible and efficient. In comparing with B/M in this category, our algorithm leads clearly for all types of matrices.

Type		n = 100		n = 200		n = 300		n = 400		n = 500	
		A	B	A	B	A	B	A	B	A	B
1	Q-LAG	1.15	0.42	4.31	1.49	9.95	3.42	18.1	6.49	31.1	11.2
	B/M	1.69	0.60	6.84	2.42	14.9	5.17	27.7	9.57	43.9	15.8
	INV	0.90	0.32	3.49	1.16	8.14	2.76	15.2	5.44	26.4	9.60
	D&C	0.54	—	3.18	—	9.36	—	21.0	—	40.4	—
	QR	1.89	—	13.8	—	44.0	—	102.	—	196.	—
2	Q-LAG	1.18	0.43	4.44	1.57	10.3	3.60	19.0	6.55	31.9	11.6
	B/M	1.73	0.60	6.50	2.36	14.9	5.22	27.4	9.60	44.1	16.1
	INV	0.92	0.31	3.50	1.22	8.17	2.84	15.6	5.31	26.4	9.58
	D&C	0.76	—	4.62	—	14.0	—	31.3	—	59.6	—
	QR	1.85	—	13.9	—	45.0	—	104.	—	198.	—
3	Q-LAG	1.26	0.42	4.74	1.61	10.7	3.67	19.8	6.79	33.0	11.8
	B/M	1.67	0.62	6.60	2.34	15.1	5.14	27.0	0.68	44.8	16.1
	INV	0.93	0.31	3.56	1.21	8.21	2.81	15.8	5.47	26.8	9.78
	D&C	0.77	—	4.92	—	14.8	—	33.6	—	63.6	—
	QR	1.83	—	13.1	—	40.8	—	93.4	—	179.	—
4	Q-LAG	1.13	0.41	4.29	1.60	9.57	3.20	16.9	5.69	26.5	8.95
	B/M	1.69	0.62	6.69	2.33	14.7	5.15	26.0	9.14	40.4	14.1
	INV	0.92	0.32	3.54	1.20	7.83	2.63	14.0	4.77	21.9	7.34
	D&C	0.51	—	3.18	—	9.32	—	21.6	—	39.8	—
	QR	1.78	—	13.9	—	45.2	—	105.	—	203.	—
5	Q-LAG	1.16	0.43	4.57	1.67	10.8	4.35	21.5	10.2	40.2	22.1
	B/M	1.67	0.58	6.56	2.38	15.6	5.93	29.8	13.2	52.7	26.6
	INV	0.90	0.32	3.66	1.31	8.89	3.60	18.1	8.93	35.0	20.2
	D&C	0.55	—	3.23	—	9.31	—	21.2	—	39.1	—
	QR	1.74	—	13.7	—	44.3	—	103.	—	197.	—
6	Q-LAG	1.10	0.37	4.02	1.33	8.73	2.86	15.2	5.07	23.5	7.76
	B/M	1.48	0.50	5.23	1.77	11.7	4.11	20.4	6.93	31.9	10.8
	INV	0.93	0.32	3.59	1.21	7.87	2.63	14.0	4.71	22.0	7.31
	D&C	0.29	—	0.79	—	1.37	—	2.05	—	2.77	—
	QR	1.73	—	12.5	—	41.8	—	98.8	—	183.	—
7	Q-LAG	1.14	0.39	4.33	1.49	9.43	3.25	16.6	5.63	25.8	8.67
	B/M	1.70	0.67	6.68	2.34	14.9	5.20	26.1	9.19	40.9	14.2
	INV	0.91	0.31	3.66	1.25	8.15	2.78	14.5	4.90	22.9	7.65
	D&C	0.70	—	2.13	—	4.09	—	7.29	—	10.5	—
	QR	2.01	—	16.3	—	53.7	—	123.	—	234.	—
8	Q-LAG	1.24	0.44	4.79	1.69	10.6	3.58	18.8	6.36	29.1	9.85
	B/M	1.71	0.66	6.67	2.41	14.9	5.14	26.0	9.08	40.5	14.1
	INV	0.91	0.31	3.60	1.25	8.08	2.71	14.4	4.87	22.3	7.62
	D&C	0.83	—	4.95	—	14.7	—	32.8	—	59.6	—
	QR	1.86	—	14.4	—	45.0	—	104.	—	200.	—
9	Q-LAG	2.21	0.54	14.0	2.40	41.5	6.34	92.5	12.7	181.	24.3
	B/M	2.48	0.65	14.9	3.01	43.6	7.51	96.4	15.0	187.	27.7
	INV	2.00	0.44	13.2	2.13	40.1	5.77	90.0	11.8	177.	22.7
	D&C	0.36	—	1.69	—	4.19	—	8.95	—	15.3	—
	QR	1.49	—	11.2	—	35.3	—	79.1	—	157.	—
10	Q-LAG	2.71	0.51	0.24	0.12	2.82	1.16	132.	18.4	28.4	8.68
	B/M	2.57	0.52	16.8	2.76	52.4	7.83	120.	17.3	229.	30.6
	INV	2.68	0.48	0.22	0.09	2.55	0.98	131.	18.1	27.7	8.38
	D&C	0.55	—	0.49	—	0.75	—	1.38	—	1.69	—
	QR	1.56	—	11.8	—	37.0	—	87.4	—	172.	—
11	Q-LAG	1.24	0.42	4.78	1.66	10.6	3.59	18.5	6.36	28.8	9.72
	B/M	1.72	0.65	6.60	2.32	14.8	5.09	26.0	9.05	40.2	14.1
	INV	0.92	0.31	3.63	1.25	8.12	2.74	14.2	5.86	22.4	7.52
	D&C	0.81	—	4.90	—	14.5	—	32.6	—	59.5	—
	QR	1.85	—	14.7	—	45.4	—	105.	—	205.	—
12	Q-LAG	0.13	0.13	0.07	0.07	1.06	0.98	7.45	6.81	0.18	0.17
	B/M	2.51	0.54	16.9	2.74	53.0	8.07	120.	17.2	229.	31.1
	INV	0.12	0.11	0.07	0.04	1.04	0.93	7.42	6.75	0.17	0.10
	D&C	0.23	—	0.52	—	0.79	—	1.51	—	1.75	—
	QR	1.38	—	9.36	—	31.1	—	67.9	—	138.	—

Table 2: Speed test on all matrices (for types 1, 2, 3, we choose $a = 2, b = 1$). The execution time in seconds in evaluating all eigenpairs is listed in column A. The execution time in seconds in evaluating only 1/3 of the largest eigenpairs is listed in column B (Only Q-LAG and B/M have this feature).

3.6.4 Accuracy test

For those matrices with known eigenvalues (type 1 to type 5), the accuracy of a method can be determined by

$$\text{direct error: } \mathcal{D} = \max_i \frac{|\tilde{\lambda}_i - \lambda_i|}{\|T\|_1}$$

where $\tilde{\lambda}_i$ is the approximation of the exact eigenvalue λ_i of T and $\|\cdot\|_1$ is the l_1 norm. For matrices of other types, the exact eigenvalues are unavailable, the accuracy of a method is determined by the residual error and the orthogonality error in the computed solution. For $T = QDQ^T$ with computed solution $\tilde{Q}\tilde{D}\tilde{Q}^T$, these errors are computed by

$$\text{residual error: } \mathcal{R} = \max_i \frac{\|(T\tilde{Q} - \tilde{Q}\tilde{D})e_i\|_2}{|\tilde{\lambda}|_{\max}}$$

$$\text{orthogonality error: } \mathcal{O} = \max_i \|(\tilde{Q}^T\tilde{Q} - I)e_i\|_2$$

If the residual and orthogonality error are small, then the error matrix $T - \tilde{Q}\tilde{D}\tilde{Q}^T$ is small [11].

The results are listed in Table 3. It appears that our algorithm Q-LAG achieves the smallest direct error on all matrices of the first 5 types. The direct error of our algorithm as well as B/M is independent of the matrix size, whereas RFQR and QR seem to have larger direct error when the matrix size becomes larger. For nearly all matrices, the eigenvectors generated by D&C suffer the least loss of orthogonality, whereas our algorithm and B/M enjoy the smallest residual errors, except on those occasions where clusters of eigenvalues exist and the inverse iteration fails to compute orthogonal vectors.

Matrix Type		n = 100			n = 200			n = 300			n = 400			n = 500		
		$\mathcal{R}$	$\mathcal{O}$	$\mathcal{D}$	$\mathcal{R}$	$\mathcal{O}$	$\mathcal{D}$	$\mathcal{R}$	$\mathcal{O}$	$\mathcal{D}$	$\mathcal{R}$	$\mathcal{O}$	$\mathcal{D}$	$\mathcal{R}$	$\mathcal{O}$	$\mathcal{D}$
1	Q-LAG	0.84	76.1	0.50	1.27	189.	1.00	1.41	152.	1.00	1.89	186.	1.00	2.16	158.	1.00
	B/M	0.84	75.7	1.00	1.27	154.	1.00	1.49	143.	1.00	1.85	145.	1.00	1.96	173.	1.00
	D&C	3.60	11.5	1.50	3.52	13.9	2.00	3.12	16.6	2.00	3.53	21.6	2.00	3.38	20.3	2.00
	RFQR	2.97	35.4	2.50	3.72	140.	4.00	2.97	121.	3.00	3.39	161.	3.00	6.27	146.	6.50
	QR	6.01	17.4	2.00	8.05	24.5	2.75	10.2	29.6	4.25	11.2	35.0	5.50	12.8	38.7	3.00
2	Q-LAG	0.71	66.7	1.00	1.20	156.	1.00	1.50	144.	1.00	1.61	191.	1.00	2.08	168.	1.00
	B/M	0.85	130.	1.00	1.16	166.	1.00	1.49	161.	1.50	1.76	189.	1.00	2.27	171.	1.50
	D&C	3.60	10.7	1.50	3.71	14.0	1.50	3.38	17.3	2.00	3.81	19.7	2.00	3.42	22.9	2.00
	RFQR	2.35	62.5	2.50	3.52	157.	3.50	3.27	177.	3.00	3.01	80.5	3.00	3.71	149.	4.00
	QR	6.15	19.0	2.25	8.46	25.6	3.00	10.3	28.4	4.06	12.0	37.9	3.69	13.0	47.1	6.13
3	Q-LAG	0.70	72.5	0.50	1.02	72.1	0.50	1.27	112.	0.50	1.50	137.	0.50	1.78	111.	0.50
	B/M	1.13	77.9	1.00	1.13	76.3	1.00	1.16	77.1	1.00	1.39	109.	1.00	1.69	160.	1.00
	D&C	3.67	12.1	1.00	3.80	14.8	1.50	3.50	18.5	1.13	3.99	20.9	2.00	3.93	24.2	1.00
	RFQR	2.21	64.5	2.00	2.65	99.3	2.50	3.29	64.7	3.00	2.79	90.4	2.50	4.23	108.	2.50
	QR	6.57	20.1	2.00	9.06	26.9	2.63	10.7	26.5	3.75	12.7	36.0	3.19	13.9	38.2	4.00
4	Q-LAG	0.71	11.3	0.16	0.83	14.4	0.08	1.81	51.2	0.85	1.44	47.7	0.09	1.13	48.0	0.51
	B/M	0.92	12.1	0.64	1.60	45.0	1.28	1.81	72.3	0.85	1.44	56.3	1.28	2.14	121.	1.02
	D&C	5.22	11.4	2.56	6.33	12.8	2.56	6.61	15.6	1.71	7.64	16.2	3.20	8.81	19.6	2.05
	RFQR	11.0	13.1	10.9	18.0	21.5	17.9	26.9	63.1	27.3	20.6	49.5	20.5	23.1	60.4	13.3
	QR	14.9	19.1	10.2	23.9	31.6	17.9	27.0	35.0	8.53	30.7	37.5	14.1	32.2	46.4	14.8
5	Q-LAG	0.48	21.1	0.10	0.53	19.1	0.05	0.66	23.9	0.52	0.68	18.8	0.52	0.97	32.5	0.52
	B/M	0.89	24.5	0.82	0.89	54.2	0.82	0.84	68.3	1.05	0.91	40.4	1.05	1.08	56.5	1.05
	D&C	4.05	12.2	1.64	3.78	16.9	1.64	2.98	13.6	3.15	3.94	17.5	3.15	3.22	16.1	3.15
	RFQR	2.91	28.1	2.87	4.94	29.0	4.92	6.61	38.8	5.76	11.5	42.0	5.76	5.76	44.8	5.76
	QR	9.06	24.7	5.74	11.1	25.8	6.55	10.8	29.6	6.29	14.4	35.9	6.29	14.6	47.3	6.29
6	Q-LAG	0.31	3.54	—	0.28	5.57	—	0.38	4.03	—	0.28	5.55	—	1.02	4.53	—
	B/M	1.06	4.05	—	0.92	5.57	—	0.85	4.53	—	1.36	5.56	—	1.02	4.55	—
	D&C	2.29	16.7	—	3.22	19.6	—	3.06	15.9	—	2.07	20.2	—	3.01	10.2	—
	RFQR	6.56	7.01	—	13.3	6.02	—	18.7	4.54	—	30.6	5.54	—	19.9	4.05	—
	QR	12.5	74.5	—	9.06	224.	—	16.3	360.	—	14.3	486.	—	12.1	632.	—
7	Q-LAG	0.77	4.85	—	0.92	10.8	—	1.56	30.0	—	1.49	30.8	—	1.43	18.7	—
	B/M	1.13	14.8	—	0.85	51.6	—	0.99	30.0	—	1.16	30.9	—	1.07	18.4	—
	D&C	7.29	26.2	—	6.23	26.7	—	5.70	13.2	—	10.0	22.3	—	7.58	13.4	—
	RFQR	10.6	5.33	—	12.3	23.8	—	24.3	14.4	—	31.1	66.6	—	24.3	23.4	—
	QR	12.0	84.4	—	20.1	180.	—	25.3	249.	—	31.0	253.	—	31.7	343.	—
8	Q-LAG	0.84	20.8	—	0.92	37.9	—	1.17	78.9	—	1.45	112.	—	1.39	127.	—
	B/M	0.92	14.6	—	0.96	37.1	—	1.36	88.1	—	1.61	104.	—	1.19	109.	—
	D&C	8.11	13.7	—	6.60	16.2	—	7.90	21.1	—	7.79	20.0	—	8.30	23.5	—
	RFQR	4.66	13.5	—	6.56	40.4	—	7.30	57.3	—	6.20	82.2	—	28.9	55.8	—
	QR	12.0	24.1	—	16.4	27.6	—	20.9	42.1	—	22.3	36.9	—	37.3	71.3	—
9	Q-LAG	1.29	9.98	—	7.44	26.9	—	24.5	5.75	—	5.44	15.4	—	97.4	45.8	—
	B/M	0.69	4.18	—	1.25	5.28	—	2.15	5.64	—	10.7	47.6	—	20.5	15.6	—
	D&C	4.26	18.2	—	6.76	30.1	—	3.94	32.3	—	5.61	28.1	—	4.74	18.0	—
	RFQR	4.46	3.56	—	4.97	7.10	—	7.13	11.2	—	9.17	9.94	—	8.10	10.9	—
	QR	4.72	42.3	—	4.82	80.6	—	8.41	123.	—	10.7	157.	—	10.6	206.	—
10	Q-LAG	1.50	213.	—	0.91	3.14	—	2.11	39.2	—	2.99	9607	—	3.33	628.	—
	B/M	0.58	257.	—	1.40	45.4	—	1.28	1800	—	2.01	484.	—	2.88	5683	—
	D&C	1.14	19.4	—	1.31	16.2	—	1.36	13.5	—	1.41	18.0	—	1.49	15.3	—
	RFQR	0.01	4.91	—	0.22	9.09	—	0.08	10.7	—	0.17	11.4	—	0.02	13.4	—
	QR	0.03	42.3	—	0.41	67.0	—	0.58	95.5	—	0.17	194.	—	0.49	210.	—
11	Q-LAG	0.45	9.57	—	0.98	28.9	—	0.60	42.5	—	0.69	83.4	—	0.83	61.8	—
	B/M	0.81	9.64	—	1.02	40.6	—	0.98	40.0	—	1.00	68.7	—	0.91	71.3	—
	D&C	4.06	13.8	—	3.62	16.5	—	10.9	23.0	—	3.29	18.1	—	4.48	24.8	—
	RFQR	8.48	11.7	—	3.09	25.8	—	10.9	23.0	—	14.1	54.9	—	16.7	35.0	—
	QR	8.63	19.7	—	9.44	29.7	—	14.5	47.6	—	18.0	53.8	—	18.0	81.7	—
12	Q-LAG	1.74	4.73	—	1.75	2.17	—	1.11	13.7	—	1.75	135.	—	3.16	3.38	—
	B/M	1.23	102.	—	6.48	3496	—	1.48	18.5	—	0.70	92.6	—	0.80	43.0	—
	D&C	0.37	17.2	—	0.28	17.4	—	0.52	14.0	—	1.01	20.0	—	1.53	11.9	—
	RFQR	0.23	5.25	—	0.28	10.2	—	0.48	12.3	—	0.02	16.1	—	0.06	14.8	—
	QR	0.45	38.6	—	0.64	80.8	—	0.49	148.	—	0.04	228.	—	0.64	358.	—

Table 3: Accuracy on all matrices (for types 1, 2, 3, we choose $a = 2, b = 1$). All errors in the table are relative to machine precision ε . Note that only the first 5 types of matrices have valid direct error $\mathcal{D}$.

4 The complex quasi-Laguerre iteration

In the rest of the work, we shall turn our attention to solving polynomial with complex zeros. Although the quasi-Laguerre iteration is derived for solving polynomials with only real zeros, it can be used to solve those with complex zeros in the same way Laguerre's iteration is used. We shall only focus on approximating simple zeros.

4.1 The formulation

Given a polynomial $f(x)$ of degree n with all of its zeros being real, the quasi-Laguerre iteration is formulated in Section 2.1 as follows: If z is a simple zero of f and is the closest zero to the right of two iterates $x_+^{(k-1)} < x_+^{(k)}$, then $x_+^{(k+1)}$, obtained from (31) with $m = 1$, is the next quasi-Laguerre iterate. Specifically, let $a = x_+^{(k-1)}$ and $b = x_+^{(k)}$, then $x_+^{(k+1)}$ is the larger solution of u of

$$\begin{aligned} \frac{1}{a-u} + \frac{n-1}{a-v} &= q(a) = \frac{f'(a)}{f(a)} \\ \frac{1}{b-u} + \frac{n-1}{b-v} &= q(b) = \frac{f'(b)}{f(b)} \end{aligned} \tag{55}$$

since it is the solution to the optimization problem (P2) with $m = 1$ in Chapter 1. Similarly, if z is the closest simple zero to the left of two iterates $x_-^{(k)} < x_-^{(k-1)}$, then $x_-^{(k+1)}$, obtained from (32) with $m = 1$, is the next quasi-Laguerre iterate. If we let $a = x_-^{(k)}$ and $b = x_-^{(k-1)}$, then $x_-^{(k+1)}$ is in fact the smaller solution of u of equations (55) since it is the solution to the optimization problem (P1) with $m = 1$. We shall use this essential property to establish the quasi-Laguerre iteration for solving polynomials with complex zeros.

Let $f(x)$ be a complex polynomial of degree n ($n \geq 3$), and z be a simple zero, real or complex, of $f(x)$. Given two complex iterates y_{k-1}

and y_k for z , the next quasi-Laguerre iterate y_{k+1} is defined to be one of the solutions of u for the following equations:

$$\begin{aligned} \frac{1}{y_k - u} + \frac{n-1}{y_k - v} &= q(y_k) = \frac{f'(y_k)}{f(y_k)} \\ \frac{1}{y_{k-1} - u} + \frac{n-1}{y_{k-1} - v} &= q(y_{k-1}) = \frac{f'(y_{k-1})}{f(y_{k-1})}. \end{aligned}$$

By solving the above equations,

$$u = y_k - \frac{2[n - (y_{k-1} - y_k)q(y_{k-1})]}{(y_{k-1} - y_k)R + 2q(y_{k-1}) \pm \sqrt{R[(y_{k-1} - y_k)^2 R + 4(n-1)]}} \quad (56)$$

where

$$R = n \left(\frac{q(y_k) - q(y_{k-1})}{y_{k-1} - y_k} \right) - q(y_k)q(y_{k-1}).$$

There are two values for u in the above formula. We will choose y_{k+1} to be the one which is closer to y_k , i.e., $|y_{k+1} - y_k|$ is smaller. More precisely, let

$$\Delta^2 = R[(y_{k-1} - y_k)^2 R + 4(n-1)] \quad \text{and} \quad \Omega = (y_{k-1} - y_k)R + 2q(y_{k-1}).$$

Since

$$|\Omega \pm \Delta|^2 = |\Omega|^2 + |\Delta|^2 \pm 2\text{Re}(\bar{\Omega}\Delta),$$

to make $|y_{k+1} - y_k|$ smaller or $|\Omega \pm \Delta|$ larger, we choose

$$y_{k+1} = \begin{cases} y_k - \frac{2[n - (y_{k-1} - y_k)q(y_{k-1})]}{\Omega + \Delta}, & \text{if } \text{Re}(\bar{\Omega}\Delta) \geq 0; \\ y_k - \frac{2[n - (y_{k-1} - y_k)q(y_{k-1})]}{\Omega - \Delta}, & \text{if } \text{Re}(\bar{\Omega}\Delta) < 0. \end{cases} \quad (57)$$

To start the iteration, two initial points are required. For given y_0 , we always apply one step of Newton's iteration at y_0 to obtain the second initial point; namely,

$$y_1 = y_0 - \frac{f(y_0)}{f'(y_0)}.$$

4.2 Convergence properties

Theorem 4.1 *The quasi-Laguerre iteration $\{y_k\}$ defined in the last section converges to z when y_0 is close to z . The ultimate convergence rate is $\sqrt{2} + 1$.*

Proof. Let

$$\epsilon_k = \frac{y_k - z}{y_{k-1} - z}, \quad \forall k \geq 1.$$

Clearly $\{y_k\}$ converges to z if for some $r < 1$, $|\epsilon_k| < r$, $\forall k$. From (56),

$$\epsilon_{k+1} = \frac{y_{k+1} - z}{y_k - z} = 1 - \frac{2[n - (y_{k-1} - y_k)q(y_{k-1})]}{(y_k - z) \left\{ (y_{k-1} - y_k)R + 2q(y_{k-1}) \pm \sqrt{R[(y_{k-1} - y_k)^2 R + 4(n-1)]} \right\}}. \quad (58)$$

Let $z^{(1)}, \dots, z^{(n-1)}$ be the zeros of f besides z . Then,

$$\begin{aligned} q(y_k) &= \frac{f'(y_k)}{f(y_k)} = \frac{1}{y_k - z} + Q(y_k) \quad \text{with} \quad Q(y) \equiv \sum_{i=1}^{n-1} \frac{1}{y - z^{(i)}}, \\ q(y_{k-1}) &= \frac{f'(y_{k-1})}{f(y_{k-1})} = \frac{1}{y_{k-1} - z} + Q(y_{k-1}) \end{aligned}$$

and

$$\begin{aligned} R &= n \left(\frac{q(y_k) - q(y_{k-1})}{y_{k-1} - y_k} \right) - q(y_k)q(y_{k-1}) \\ &= \frac{n-1}{(y_k - z)(y_{k-1} - z)} - \frac{Q(y_{k-1})}{y_k - z} - \frac{Q(y_k)}{y_{k-1} - z} - Q(y_k)Q(y_{k-1}) \\ &\quad + nR_1(y_{k-1}, y_k) \quad \text{with} \quad R_1(y_{k-1}, y_k) \equiv \sum_{i=1}^{n-1} \frac{1}{(y_{k-1} - z^{(i)})(y_k - z^{(i)})}. \end{aligned}$$

Now,

$$\begin{aligned} (a) \quad (y_{k-1} - y_k)q(y_{k-1}) &= (y_{k-1} - y_k) \left(\frac{1}{y_{k-1} - z} + Q(y_{k-1}) \right) \\ &= (1 - \epsilon_k) + Q(y_k)(1 - \epsilon_k)(y_{k-1} - z) = 1 - \epsilon_k + O(y_{k-1} - z) \\ (b) \quad (y_k - z)(y_{k-1} - z)R &= n - 1 - (y_{k-1} - z)Q(y_{k-1}) \end{aligned}$$

$$\begin{aligned}
& -(y_k - z)Q(y_k) - Q(y_k)Q(y_{k-1})(y_k - z)(y_{k-1} - z) \\
& + n(y_k - z)(y_{k-1} - z)R_1(y_{k-1}, y_k) = n - 1 + O(y_{k-1} - z) \\
(c) \quad & (y_k - z)(y_{k-1} - y_k)R = (1 - \epsilon_k)(y_k - z)(y_{k-1} - z)R \\
& = (1 - \epsilon_k)(n - 1) + O(y_{k-1} - z) \quad (\text{because of (b) above}) \\
(d) \quad & (y_k - z)^2 R = \epsilon_k(y_k - z)(y_{k-1} - z)R \\
& = \epsilon_k(n - 1) + O(y_{k-1} - z) \quad (\text{because of (b) above}) \\
(e) \quad & (y_k - z)^2 R [(y_{k-1} - y_k)^2 R + 4(n - 1)] \\
& = [(y_k - z)(y_{k-1} - y_k)R]^2 + 4(n - 1)(y_k - z)^2 R \\
& = (1 - \epsilon_k)^2(n - 1)^2 + 4(n - 1)\epsilon_k(n - 1) + O(y_{k-1} - z) \\
& = (n - 1)^2(1 + \epsilon_k)^2 + O(y_{k-1} - z) \\
& \quad (\text{because of (c) and (d) above}).
\end{aligned}$$

If $y_0 \rightarrow z$ and $y_1 \rightarrow z$, then

$$\begin{aligned}
Q(y_0) &= \sum_{i=1}^{n-1} \frac{1}{y_0 - z^{(i)}} \rightarrow Q \equiv \sum_{i=1}^{n-1} \frac{1}{z - z^{(i)}} < \infty, \\
R_1(y_0, y_1) &= \sum_{i=1}^{n-1} \frac{1}{(y_0 - z^{(i)})(y_1 - z^{(i)})} \rightarrow R^* \equiv \sum_{i=1}^{n-1} \frac{1}{(z - z^{(i)})^2} < \infty.
\end{aligned}$$

In particular, if y_0 is close to z and $|\epsilon_1| < 1$, then $|y_1 - z| < |y_0 - z|$ and

$$|Q(y_0)| < 2|Q| \quad \text{and} \quad |R_1(y_0, y_1)| < 2|R^*|.$$

In addition, if $|\epsilon_i| < 1$ for $i = 2, \dots, k$, then

$$|y_k - z| < \dots < |y_1 - z| < |y_0 - z|.$$

Hence,

$$|Q(y_k)| < 2|Q| \quad \text{and} \quad |R_1(y_{k-1}, y_k)| < 2|R^*|.$$

It follows that the bounds of $O(y_{k-1} - z)$ in all the above expressions (a) through (e) are independent of k . Substituting (a), (c), (e) into (58)

yields

$$\epsilon_{k+1} = 1 - \frac{2(n-1) + 2\epsilon_k + \alpha_1}{(n-1)(1-\epsilon_k) + 2\epsilon_k \pm (n-1)(1+\epsilon_k) + \alpha_2},$$

where $\alpha_1 = O(y_{k-1} - z)$ and $\alpha_2 = O(y_{k-1} - z)$; namely, $\exists C_1 > 0$ and $C_2 > 0$, independent of k , such that

$$|\alpha_1| \leq C_1 |y_{k-1} - z| \quad \text{and} \quad |\alpha_2| \leq C_2 |y_{k-1} - z|.$$

Let

$$\begin{aligned} A &= (n-1)(1-\epsilon_k) + 2\epsilon_k \pm (n-1)(1+\epsilon_k) \\ &= \begin{cases} 2(n-1) + 2\epsilon_k & \text{if '+' is chosen;} \\ (4-2n)\epsilon_k & \text{if '-' is chosen.} \end{cases} \end{aligned}$$

In (57), since

$$(y_k - z)(\Omega \pm \Delta) = A + \alpha_2,$$

thus, making $|y_{k+1} - y_k|$ smaller is equivalent to making $|A|$ larger. When $|\epsilon_k| < 1$, $|2(n-1) + 2\epsilon_k| > |(4-2n)\epsilon_k|$. Hence, to have $|A|$ larger, we always choose '+' for A , making

$$A = 2(n-1) + 2\epsilon_k$$

and

$$|A| = |2(n-1) + 2\epsilon_k| \geq 2(n-1) - 2|\epsilon_k| \geq 2n - 4.$$

So,

$$\epsilon_{k+1} = 1 - \frac{A + \alpha_1}{A + \alpha_2} = \frac{\alpha_2 - \alpha_1}{A + \alpha_2}. \quad (59)$$

If $|\epsilon_1| < 1$, $|\epsilon_2| < 1, \dots, |\epsilon_k| < 1$ and $|y_0 - z| < \frac{1}{C_2}$, then

$$|\alpha_2| \leq C_2 |y_{k-1} - z| < C_2 |y_0 - z| < 1.$$

Hence,

$$|A + \alpha_2| \geq |A| - |\alpha_2| \geq 2n - 5 > 0.$$

From (59),

$$|\epsilon_{k+1}| = \frac{|\alpha_2 - \alpha_1|}{|A + \alpha_2|} \leq \left(\frac{C_1 + C_2}{2n - 5} \right) |y_{k-1} - z|.$$

In summary, we have shown the following:

Lemma 4.2 *If y_0 is close to z , then there exists a constant $C > 0$ such that $\forall k \geq 1$, if $|\epsilon_1| < 1, \dots, |\epsilon_k| < 1$, then*

$$|\epsilon_{k+1}| \leq C|y_{k-1} - z|. \quad (60)$$

Now, given a real number $r < 1$, when $|y_0 - z| < \frac{r}{C}$, by (60) above,

$$|\epsilon_2| \leq C|y_0 - z| < r \quad \text{or} \quad |y_2 - z| < r|y_1 - z|.$$

Notice that from the way y_1 is chosen; namely, by applying one step of Newton's iteration on given y_0 , we always have

$$|y_1 - z| < |y_0 - z|, \quad \text{or equivalently} \quad |\epsilon_1| < 1$$

when y_0 is close to z , say for instance, y_0 is in the convergence neighborhood of Newton's iteration at z . By (60) again,

$$|\epsilon_3| \leq C|y_1 - z| < C|y_0 - z| < r, \quad \text{and so,} \quad |y_3 - z| < r|y_2 - z|.$$

It follows recursively that

$$|\epsilon_{k+1}| \leq C|y_{k-1} - z| < r \quad \text{and} \quad |y_{k+1} - z| < r|y_k - z|, \quad \forall k \geq 1$$

since C is independent of k . Therefore $\{y_k\}$ converges to z . Furthermore, from (60),

$$|\epsilon_{k+1}| \leq C|y_{k-1} - z| \rightarrow 0, \quad \text{as } k \rightarrow \infty.$$

Thus the convergence of $\{y_k\}_{k=1}^{\infty}$ to z is super-linear.

The rest of the proof follows exactly the same argument as the proof of part (ii) of Theorem 2.3 with $m = M = 1$. $\square$

References

- [1] E. ANDERSON, Z. BAI, C. BISCHOF, J. DEMMEL, J. DONGARRA, J. DU CROZ, A. GREENBAUM, S. HAMMARLING, A. MCKENNEY, S. OSTROUCHOV, and D. SORENSON, *LAPACK User's Guide*, SIAM, Philadelphia, 1992.
- [2] J. J. M. CUPPEN, *A divide and conquer method for the symmetric tridiagonal eigenproblem*, Numer. Math., 36 (1981), pp. 177–195.
- [3] J. J. DONGARRA AND D. C. SORENSEN, *A fully parallel algorithm for the symmetric eigenvalue problem*, SIAM J. Sci. Stat. Comput., 8 (1987), pp. 139–154.
- [4] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, 2nd Ed., The Johns Hopkins University Press, Baltimore, MD, 1989.
- [5] R. T. GREGORY AND D. L. KARNEY, *A Collection of Matrices for Testing Computational Algorithms*, Robert E. Krieger Publishing Company, Huntington, New York, 1978.
- [6] M. GU AND S. C. EISENSTAT, *A divide-and-conquer algorithm for the symmetric tridiagonal eigenproblem*, SIAM J. Matrix Anal. Appl., Vol. 16, No. 1 (1995), pp. 172–191.
- [7] W. KAHAN, *Notes On Laguerre's Iteration*, preprint, University of California, Berkeley (1992).
- [8] T. Y. LI AND Z. ZENG, *Laguerre's iteration in solving the symmetric tridiagonal eigenproblem — revisited*, SIAM J. Sci. Comput., Vol. 15, No. 5 (1994), pp. 1145–1173.
- [9] B. N. PARLETT, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, NJ, 1980.
- [10] B. N. PARLETT, *The use of a refined error bound when updating eigenvalues of tridiagonals*, Lin. Alg. & Appls., Vol. 68 (1985), pp. 179–219.
- [11] J. RUTTER, *A serial implementation of Cuppen's divide and conquer algorithm for the symmetric eigenvalue problem*, LAPACK lawn 69 (1994).

- [12] D. C. SORENSEN AND P. T. P. TANG, *On the orthogonality of eigenvectors computed by divide-and-conquer techniques*, SIAM. J. Numer. Anal., 28 (1991), pp. 1752 - 1775.
- [13] C. TREFFTZ, C. C. HUANG, P. MCKINLEY, T. Y. LI, AND Z. ZENG, *A scalable eigenvalue solver for symmetric tridiagonal matrices*, to appear, Parallel Comput.
- [14] J. H. WILKINSON, *The Algebraic Eigenvalue Problem*, Oxford University Press, Oxford, 1965.