

Algorithms For Inverse Eigenvalue Problems†

REN-CANG LI*

(*Computing Center, Academia Sinica*)

ABSTRACT

Two new algorithms based on QR decompositions (QRDs) (with column pivoting) are proposed for solving inverse eigenvalue problems, and under some non-singularity assumptions they are both locally quadratically convergent.

Several numerical tests are presented to illustrate their convergence behavior.

§1. INTRODUCTION

Inverse eigenvalue problems arise often in applied mathematics (see Friedland et al [1] §1), and they have been treated by many mathematicians. Some of these papers are among the references. Let A be a fixed $n \times n$ (complex) valued matrix. The most common inverse eigenvalue problems are the two following problems proposed by Downing and Householder [11]:

(i) Find a diagonal complex valued matrix D such that the spectrum of $A + D$ is a given set $\lambda^* = (\lambda_1^*, \dots, \lambda_n^*)$.

(ii) Find a diagonal complex valued matrix D such that the spectrum of AD is a given set $\lambda^* = (\lambda_1^*, \dots, \lambda_n^*)$.

The first problem is called *the inverse additive eigenvalue problem*, and the second one *the multiplicative eigenvalue problem*. Often in practical applications, A , D and λ^* in the above two problems are real.

NOTATION: We shall use $\mathbb{C}^{m \times n}$ ($\mathbb{R}^{m \times n}$) for the m by n complex (real) matrix set, $\mathbb{C}^m = \mathbb{C}^{m \times 1}$ ($\mathbb{R}^m = \mathbb{R}^{m \times 1}$), $\mathbb{C} = \mathbb{C}^1$ ($\mathbb{R} = \mathbb{R}^1$); $\mathcal{U}_n \subset \mathbb{C}^{n \times n}$ denotes the n by n unitary matrix set. $I^{(n)}$ is the n by n unit matrix, $e_j^{(n)}$ the j th column of $I^{(n)}$ and

$$I_j^{(n)} \equiv (e_1^{(n)}, \dots, e_j^{(n)}),$$

when no confusion arises, these superscripts (n) are usually omitted. A^H , A^T denote the conjugate transpose and transpose respectively, and $\|A\|_2$ the spectral norm of A .

For convenience, in this paper we generalize the statement of the above two inverse problems as

†Published in *J. Comp. Math.*, **10**(1992), 97–111.

*Current address: Institute of Applied Physics and Computational Mathematics, P. O. Box 8009, Beijing 100088, P. R. China.

PROBLEM G. Let $A(c) \in \mathbb{C}^{n \times n}$ be a differentiable matrix-valued function of $c \in \mathbb{C}^n$. Find a point $c^* \in \mathbb{C}^n$ such that the spectrum of matrix $A(c^*)$ is a given set $\lambda^* = (\lambda_1^*, \dots, \lambda_n^*)$.

Here, that we say $A(c) \in \mathbb{C}^{n \times n}$ is differentiable with respect to c means for any $c^{(0)} \in \mathbb{C}^n$, we have

$$A(c) = A(c^{(0)}) + \sum_{i=1}^n \frac{\partial}{\partial c_i} A(c) \Big|_{c=c^{(0)}} (c_i - c_i^{(0)}) + o(\|c - c^{(0)}\|_2), \quad (1.1)$$

where

$$c = (c_1, \dots, c_n)^T, \quad c^{(0)} = (c_1^{(0)}, \dots, c_n^{(0)})^T, \quad (1.2)$$

$$\frac{\partial}{\partial c_i} A(c) = \left(\frac{\partial}{\partial c_i} a_{kj}(c) \right) \in \mathbb{C}^{n \times n} \quad \text{for} \quad A(c) = (a_{kj}(c))$$

and

$$\|c - c^{(0)}\|_2 = \left(\sum_{i=1}^n |c_i - c_i^{(0)}|^2 \right)^{\frac{1}{2}}. \quad (1.3)$$

As to the solvability, and some numerical methods of inverse additive, multiplicative eigenvalue problems, we refer readers to e.g. Sun [7], Friedland et al [1]-[3] and other related references therein. The aims of this paper is to propose two new methods to solve Problem G and to analyse their convergence behavior at appropriate hypotheses. Throughout this paper we assume the given set λ^* satisfy

$$\lambda_i^* \neq \lambda_j^* \quad \text{for} \quad i \neq j, \quad (1.4)$$

and Problem G itself is solvable.

The rest of this paper is organized as follows: In §2 we cite some necessary differentiability theorems proved in Li [12]. In §3 we first discuss some formulations of numerical methods, then give our algorithms and their convergence analysis and finally in §4 we present several numerical tests to illustrate their behavior.

§2. QR DECOMPOSITION(QRD) AND DIFFERENTIABILITY

Let $A(c) \in \mathbb{C}^{n \times n}$, the QRDs with column pivoting (see Golub & van Loan [9, pp.163-167]) of $A(c)$ can be read as

$$A(c)\pi(c) = Q(c)R(c), \quad Q(c) \in \mathcal{U}_n, \quad (2.1)$$

where $\pi(c) \in \mathbb{C}^{n \times n}$ is a permutation matrix and $R(c) \in \mathbb{C}^{n \times n}$ an upper triangular matrix. The following theorem was obtained by [12], and it is the base of this paper.

THEOREM 2.1. Let $A(c) \in \mathbb{C}^{n \times n}$ be a matrix-valued function of $c \in \mathbb{C}^n$, $\pi \in \mathbb{C}^{n \times n}$ a permutation matrix and $c^{(0)} \in \mathbb{C}^n$. Assume that the first $n - 1$ column vectors of $A(c^{(0)})\pi$ are linearly independent, and

$$A(c^{(0)})\pi = Q_0 R_0$$

is its any QRD. If $A(c)$ is differentiable at $c^{(0)}$ (refer to (1.1)), then there exists a neighborhood $\mathcal{B}(c^{(0)}) \subset \mathbb{C}^n$ of $c^{(0)}$ such that we have QRDs of $A(c)\pi$

$$A(c)\pi = Q(c)R(c) \quad \text{for } c \in \mathcal{B}(c^{(0)}) \tag{2.2}$$

satisfying that

$$Q(c^{(0)}) = Q_0 \quad \text{and} \quad R(c^{(0)}) = R_0,$$

and that the diagonal elements of $R(c)$ are differentiable at $c^{(0)}$,

$$\begin{aligned} r_{nn}(c) &\equiv e_n^T R(c) e_n \\ &= e_n^T R_0 e_n + \sum_{i=1}^n \left[e_n^T Q_0^H \frac{\partial}{\partial c_i} A(c) \Big|_{c=c^{(0)}} \pi e_n \right. \\ &\quad \left. - e_n^T Q_0^H \frac{\partial}{\partial c_i} A(c) \Big|_{c=c^{(0)}} \pi I_{n-1} (I_{n-1}^T R_0 I_{n-1})^{-1} I_{n-1}^T R_0 e_n \right] (c_i - c_i^{(0)}) \\ &\quad + o(\|c - c^{(0)}\|_2), \end{aligned} \tag{2.3}$$

where $c, c^{(0)}$ are of forms (1.2). Moreover if $A(c^{(0)})$ is second order differentiable, the last term $o(\|c - c^{(0)}\|_2)$ in the expansion (1.1) of $A(c)$ can be replaced by $O(\|c - c^{(0)}\|_2^2)$, the last term in (2.3) can also be improved as $O(\|c - c^{(0)}\|_2^2)$.

The original form of Theorem 2.1 in [12] is slightly different from that given here. The difference lies in that here we point out explicitly arbitrarily choosing a QRD of $A(c^{(0)})\pi$, which is implied in the proof of Theorem 2.1 in [12], and we must bear in mind that QRDs (2.2) vary with this initial QRD.

§3. FORMULATIONS OF NUMERICAL METHODS AND ALGORITHMS

Once again, we emphasize that the hypothesis (1.4) is always satisfied in this paper.

3.1 Formulations Of Numerical Methods. So far, existing algorithms which are locally quadratically convergent (under appropriate conditions) were obtained by first constructing a nonlinear system and then applying Newton method to it. Practical applications have shown that solving general inverse eigenvalue problems such as Problem G in such a way is feasible. The following are the two formulations of the problems.

FORMULATION I. Suppose $A(c) \in \mathbb{C}^{n \times n}$ is real symmetric or Hermitian, and its n eigenvalues are arranged in ascending order

$$\lambda_1(c) \leq \dots \leq \lambda_n(c).$$

Solve the nonlinear system

$$f(c) = \begin{pmatrix} \lambda_1(c) - \lambda_1^* \\ \vdots \\ \lambda_n(c) - \lambda_n^* \end{pmatrix} = 0, \tag{3.1}$$

here we assume $\lambda_1^* < \dots < \lambda_n^*$.

FORMULATION II. Solve the nonlinear system

$$g(c) = \begin{pmatrix} \det(A(c) - \lambda_1^* I) \\ \vdots \\ \det(A(c) - \lambda_n^* I) \end{pmatrix} = 0. \quad (3.2)$$

The first formulation is the most natural one and has been used by [11] in the case of the additive eigenvalue problems (for detail on this formulation see [1], [10]). The second formulation has been proposed by [4]. Recently Ye [8] proposed two algorithms, which are proved to be linearly convergent, to solve inverse additive, multiplicative eigenvalue problems for real symmetric (Hermitian) matrices.

In this paper, we shall propose a class of algorithms based on QRDs (with column pivoting). The original idea of these algorithms comes from Kublanovskaya's approach [13] for solving nonlinear eigenvalue problems. We remark here that nonlinear eigenvalue problems may be regarded as a special kind inverse eigenvalue problems since finding a $\lambda \in \mathbb{C}$ such that

$$\det B(\lambda) = 0 \quad \text{for } B(\lambda) \in \mathbb{C}^{n \times n}$$

may be thought of as finding a $\lambda \in \mathbb{C}$ such that zero is an eigenvalue of $B(\lambda)$. In such inverse problems the specified eigenvalue(s) are zeros.

Now, we are in a position to give our formulation of the equivalent nonlinear system to Problem G. Compute QRDs (with column pivoting) of $A(c) - \lambda_i^* I$ ($i = 1, \dots, n$)

$$(A(c) - \lambda_i^* I)\pi_i(c) = Q_i(c)R_i(c), \quad i = 1, \dots, n, \quad (3.3)$$

and assume permutation matrices $\pi_i(c) \in \mathbb{C}^{n \times n}$ are constant in a sufficiently small neighborhood of c for each i . From Theorem 2.1, we can make that n functions

$$e_n^T R_i(c) e_n, \quad i = 1, \dots, n \quad (3.4)$$

are differentiable at $c \in \mathbb{C}^n$. If column pivoting is performed then we have

$$|e_1^T R_i(c) e_1| \geq |e_2^T R_i(c) e_2| \geq \dots \geq |e_n^T R_i(c) e_n|, \quad i = 1, \dots, n, \quad (3.5a)$$

or any other methods are used such that

$$|e_j^T R_i(c) e_j| \geq |e_n^T R_i(c) e_n|, \quad j = 1, \dots, n-1. \quad (3.5b)$$

Therefore the spectrum of $A(c)$ is λ^* , if and only if

$$e_n^T R_i(c) e_n = 0, \quad i = 1, \dots, n. \quad (3.6)$$

Hence we introduce

FORMULATION III. *Solve the nonlinear system*

$$h(c) = \begin{pmatrix} e_n^T R_1(c) e_n \\ \vdots \\ e_n^T R_n(c) e_n \end{pmatrix} = 0. \tag{3.7}$$

It is worth mentioning that because of the non-uniqueness of QRDs $h(c)$ is not uniquely determined for any c . However we remark that such “flexiability” doesn’t affect the effectiveness of our algorithms; and this is our mainly concern (refer to Lemma 3.2 below).

3.2. Algorithms. The following algorithm follows straightforwardly from Theorem 2.1, Formulation III and Newton method.

ALGORITHM 3.1: Find a solution $c^* \in \mathbb{C}^n$ to the solvable Problem G.

- a) Given an initial approximation $c^{(0)}$ of c^* .
- b) Compute

$$B^{(i,\nu)} \equiv A(c^{(\nu)}) - \lambda_i^* I, \quad i = 1, \dots, n,$$

$$A_1^{(k,\nu)} \equiv \left. \frac{\partial}{\partial c_k} A(c) \right|_{c=c^{(\nu)}}, \quad k = 1, \dots, n.$$

- c) Compute QRDs with column pivoting of $B^{(i,\nu)}$ ($i = 1, \dots, n$)

$$B^{(i,\nu)} \pi^{(i,\nu)} = Q^{(i,\nu)} R^{(i,\nu)}, \quad R^{(i,\nu)} = \begin{pmatrix} R_{11}^{(i,\nu)} & R_{12}^{(i,\nu)} \\ 0 & r_{nn}^{(i,\nu)} \end{pmatrix}.$$

- d) Compute for $i, k = 1, \dots, n$

$$j_{ik}^{(\nu)} = e_n^T Q^{(i,\nu)H} A_1^{(k,\nu)} \pi^{(i,\nu)} e_n - e_n^T Q^{(i,\nu)H} A_1^{(k,\nu)} \pi^{(i,\nu)} I_{n-1} R_{11}^{(i,\nu)-1} R_{12}^{(i,\nu)},$$

and let $J^{(\nu)} = (j_{ik}^{(\nu)})$.

- e) Solve linear systems

$$J^{(\nu)}(c^{(\nu+1)} - c^{(\nu)}) = -b^{(\nu)},$$

where $b^{(\nu)} = (r_{nn}^{(1,\nu)}, \dots, r_{nn}^{(n,\nu)})^T \in \mathbb{C}^n$.

- f) If the needed accuracy is attended, **stop**; otherwise go to b).

Now we give a rough estimation of the computational requirements of Algorithm 3.1. Since in all Newton kind algorithms for solving Problem G, Steps b) and e) are indispensable, therefore our estimation do not include the computational requirements for Steps b) and e). Assume that $A(c)$ is dense. It is well-known that the QRDs obtained by using Householder transformations for each $B^{(i,\nu)}$ requires approximately $\frac{2}{3}n^3$ multiplications (e.g. see [9] p.165), where $Q^{(i,\nu)}$ is stored in factor forms. Therefore Step c) requires approximately $\frac{2}{3}n^4$ multiplications. For Step d), one can easily verify that it requires approximately n^4 multiplications. Thus Algorithm 3.1 requires approximately $\frac{5}{3}n^4$ multiplications per iteration.

it follows that

$$Q_1^H A \pi = R_1 \pi = \begin{pmatrix} r_{11}^{(1)} & \cdots & r_{1l-1}^{(1)} & r_{1l+1}^{(1)} & \cdots & r_{1n}^{(1)} & r_{1l}^{(1)} \\ & \ddots & \vdots & \vdots & & \vdots & \vdots \\ & & r_{l-1l-1}^{(1)} & r_{l-1l+1}^{(1)} & \cdots & r_{l-1n}^{(1)} & r_{l-1l}^{(1)} \\ & & & r_{ll+1}^{(1)} & \cdots & r_{ln}^{(1)} & r_{ll}^{(1)} \\ & & & r_{l+1l+1}^{(1)} & \ddots & \vdots & 0 \\ & & & & \ddots & r_{n-1n}^{(1)} & \vdots \\ & & & & & r_{nn}^{(1)} & 0 \end{pmatrix}.$$

Now similarly as before, we choose $n - l$ Givens rotations $G(l, l + 1), \dots, G(n - 1, n)$ such that

$$\begin{aligned} Q_2^H Q_1^H A \pi &\equiv G(n - 1, n) \cdots G(l, l + 1) Q_1^H A \pi \\ &= G(n - 1, n) \cdots G(l, l + 1) R_1 \pi \\ &= R \end{aligned} \tag{3.10}$$

is an upper triangular matrix. It is easily verify that ($R = (r_{ij})$)

$$r_{ii} \begin{cases} = r_{ii}^{(1)}, & 1 \leq i \leq l - 1, \\ \geq r_{i+1i+1}^{(1)}, & l \leq i \leq n - 1, \\ \leq r_{ll}^{(1)}, & i = n. \end{cases}$$

Therefore

$$|e_n^T R e_n| = |r_{nn}| \leq |e_j^T R e_j|, \quad j = 1, \dots, n - 1,$$

which is just (3.5b). Hence $Q = Q_1 Q_2, \pi$ and R as shown by (3.9), (3.10) respectively meet our needs.

ALGORITHM 3.2: Find a solution $c^* \in \mathbb{C}^n$ to the solvable Problem G.

- a) Given an initial approximation $c^{(0)}$ of c^* .
- b) Compute

$$\begin{aligned} B^{(i,\nu)} &\equiv A(c^{(\nu)}) - \lambda_i^* I, \quad i = 1, \dots, n, \\ A_1^{(k,\nu)} &\equiv \left. \frac{\partial}{\partial c_k} A(c) \right|_{c=c^{(\nu)}}, \quad k = 1, \dots, n. \end{aligned}$$

- c) Find factorizations

$$U^{(\nu)H} A(c^{(\nu)}) U^{(\nu)} = H^{(\nu)}, \quad U^{(\nu)} \in \mathcal{U}_n,$$

where $H^{(\nu)}$ are Hessenberg matrices, and compute $B^{(i,\nu)} = H^{(\nu)} - \lambda_i^* I (1 \leq i \leq n)$ which are also Hessenberg matrices. Use the techniques discribed above to give QRDs of $B^{(i,\nu)}$ ($i = 1, \dots, n$)

$$B^{(i,\nu)} \pi^{(i,\nu)} = Q^{(i,\nu)} R^{(i,\nu)}, \quad R^{(i,\nu)} = \begin{pmatrix} R_{11}^{(i,\nu)} & R_{12}^{(i,\nu)} \\ 0 & r_{nn}^{(i,\nu)} \end{pmatrix}.$$

d) Compute for $i, k = 1, \dots, n$,

$$j_{ik}^{(\nu)} = e_n^T Q^{(i,\nu)H} U^{(\nu)H} A_1^{(k,\nu)} U^{(\nu)} \pi^{(i,\nu)} e_n \\ - e_n^T Q^{(i,\nu)H} U^{(\nu)H} A_1^{(k,\nu)} U^{(\nu)} \pi^{(i,\nu)} I_{n-1} R_{11}^{(i,\nu)-1} R_{12}^{(i,\nu)},$$

and let $J^{(\nu)} = (j_{ik}^{(\nu)})$.

e) Solve linear systems

$$J^{(\nu)}(c^{(\nu+1)} - c^{(\nu)}) = -b^{(\nu)},$$

where $b^{(\nu)} = (r_{nn}^{(1,\nu)}, \dots, r_{nn}^{(n,\nu)})^T \in \mathbb{C}^n$.

f) If the needed accuracy is attended, **stop**; otherwise go to b).

As to the computational requirements of Algorithm 3.2 we note that Step c) requires approximately $O(n^3)$ multiplications and Step d) requires approximately n^4 multiplications. Thus generally Algorithm 3.2 is cheaper than Algorithm 3.1. On the other hand if $A_1^{(i,\nu)}$ is sufficiently sparse (e.g. for inverse additive, multiplicative eigenvalue problems) Step d) in Algorithm 3.2 requires approximately only $O(n^3)$ multiplications, then at this time Algorithm 3.2 requires approximately $O(n^3)$ multiplications per iteration while Algorithm 3.1 requires $O(n^4)$ multiplications.

It seems that generally the convergence domain of Algorithm 3.2 was smaller than that of Algorithm 3.1. A heuristic interpretation is that QRDs with column pivoting often reveal singularity of a matrix better than those without column pivoting. Our numerical examples in §4 also confirm such observation.

3.3. Convergence Analysis. We study first the locally quadratic convergence of Algorithm 3.1.

LEMMA 3.1([12]). *Suppose the first $n - 1$ column vectors of matrix $C \in \mathbb{C}^{n \times n}$ are linearly independent, and*

$$C = Q_1 R_1 = Q_2 R_2$$

are its two QRDs. Then there exists a diagonal matrix $D \in \mathcal{U}_n$ such that

$$Q_1 = Q_2 D \quad \text{and} \quad R_1 = D^H R_2.$$

Although the QRDs of a given matrix is not unique (neither are matrix $J^{(\nu)}$ and vector $b^{(\nu)}$ for fixed ν in Algorithm 3.1), we have

LEMMA 3.2. *In Algorithm 3.1, for any fixed ν suppose*

$$B^{(i,\nu)}\pi^{(i,\nu)} = Q_l^{(i,\nu)}R_l^{(i,\nu)}, l = 1, 2 \tag{3.11}$$

are two (different) QRDs of $B^{(i,\nu)}\pi^{(i,\nu)}$ ($i = 1, \dots, n$), and using $J_l^{(\nu)}$ and $b_l^{(\nu)}$ to denote two matrices and vectors obtained by Step d) of Algorithm 3.1 corresponding to (3.11). Then there exists a diagonal matrix $D \in \mathcal{U}_n$ such that

$$J_1^{(\nu)} = DJ_2^{(\nu)}, \quad b_1^{(\nu)} = Db_2^{(\nu)}. \tag{3.12a}$$

If moreover $J_l^{(\nu)}$ invertible, then

$$J_1^{(\nu)-1}b_1^{(\nu)} = J_2^{(\nu)-1}b_2^{(\nu)}. \tag{3.12b}$$

The proof of Lemma 3.2 is the combination of Lemma 3.1 and Step d) in Algorithm 3.1. (3.12a) characterizes the variations of $J^{(\nu)}$ and $b^{(\nu)}$ with QRDs, and (3.12b) says that given an approximation c of c^* , the improved approximation \tilde{c} obtained by performing one iteration of Algorithm 3.1 is independent of QRDs.

LEMMA 3.3([12]). *Suppose the first $n - 1$ column vectors of matrix $C_1 \in \mathbb{C}^{n \times n}$ are linearly independent, $C_1 = Q_1R_1$ is a QRD of C_1 . Let $C_2 \in \mathbb{C}^{n \times n}$. Then for any $\epsilon > 0$, there exists a QRD $C_2 = Q_2R_2$ such that*

$$\|Q_1 - Q_2\|_2 < \epsilon, \|R_1 - R_2\|_2 < \epsilon,$$

if $\|C_1 - C_2\|_2$ sufficiently small.

THEOREM 3.1. *In Algorithm 3.1, suppose $\pi^{(i,\nu)} = \pi^{(i,*)}$ is independent of ν when $\|c^{(\nu)} - c^*\|_2$ sufficiently small, and $J^* \in \mathbb{C}^{n \times n}$ corresponding to QRDs of $B^{(i,*)}\pi^{(i,*)}$ is invertible, where $B^{(i,*)} = A(c^*) - \lambda_i^*I$ ($1 \leq i \leq n$). Suppose also $A(c)$ is second order differentiable. Then, there exists $\epsilon > 0$ such that when $\|c^{(\nu)} - c^*\|_2 < \epsilon$ we have*

$$\|c^{(\nu+1)} - c^*\|_2 = O(\|c^{(\nu)} - c^*\|_2^2). \tag{3.13}$$

PROOF: $\|c^{(\nu)} - c^*\|_2$ is sufficiently small, so is $\|A(c^{(\nu)}) - A(c^*)\|_2$. Therefore from Lemma 3.2, Lemma 3.3 and the definition of $J^{(\nu)}$, we know that $J^{(\nu)}$ invertible and

$$\|J^{(\nu)-1}\|_2 \leq \|J^{*-1}\|_2 + f(\epsilon)$$

for sufficiently small ϵ , here $f(\epsilon) \geq 0$ is a continuous function of ϵ and $f(0) = 0$. Thus we have

$$\begin{aligned} \|b^{(\nu+1)}\|_2 &= \|b^{(\nu)} + J^{(\nu)}(c^{(\nu+1)} - c^{(\nu)}) + O(\|c^{(\nu+1)} - c^{(\nu)}\|_2^2)\|_2 \\ &= O(\|c^{(\nu+1)} - c^{(\nu)}\|_2^2) \\ &= O(\|J^{(\nu)-1}b^{(\nu)}\|_2^2) \\ &= O(\|b^{(\nu)}\|_2^2). \end{aligned} \tag{3.14}$$

On the other hand, it follows from linear independence of the first $n - 1$ column vectors of each $B^{(i,\nu)}\pi^{(i,\nu)}$ (since $\lambda_i^* \neq \lambda_j^*$ for $i \neq j$) and Theorem 2.1 that

$$\|b^{(\nu)}\|_2 = \|b^* + J^*(c^{(\nu)} - c^*) + O(\|c^{(\nu)} - c^*\|_2^2)\|_2 = O(\|c^{(\nu)} - c^*\|_2) \quad (3.15)$$

and similarly

$$\|b^{(\nu+1)}\|_2 = O(\|c^{(\nu+1)} - c^*\|_2), \quad (3.16)$$

where b^* is defined similarly to J^* . Combining (3.14)-(3.16) will lead to (3.13). \blacksquare

Before going into the convergence analysis of Algorithm 3.2, we remark that giving a convergence analysis of Algorithm 3.2 is much more complicated. This is because of the non-uniqueness of factorizations

$$U(c)^H A(c) U(c) = \text{upper Hessenberg matrices.} \quad (3.17)$$

At this time we even have no similar result like Lemma 3.1. In order to give an analysis, we make following additional hypothesis.

HYPOTHESIS S. *In some neighborhood of c^* , the choosing factorizations (3.17) in Algorithm 3.2 satisfy that $U(c)$ is determined uniquely by c and continuous with respect to c .*

THEOREM 3.2. *In Algorithm 3.2, suppose Hypothesis S is satisfied, and $\pi^{(i,\nu)} = \pi^{(i,*)}$ is independent of ν when $\|c^{(\nu)} - c^*\|_2$ sufficiently small, $J^* \in \mathbb{C}^{n \times n}$ corresponding to c^* is invertible. Assume also that $A(c)$ is second order differentiable. Then Algorithm 3.2 is locally quadratically convergent.*

PROOF: From Hypothesis S and Lemmas 3.1 and 3.2, we know that there exists $\epsilon_1 > 0$ such that if $\|c - c^*\|_2 < \epsilon_1$ then

$$A(c) = A(c^*) + \sum_{i=1}^n \frac{\partial}{\partial c_i} A(c) \Big|_{c=c^*} (c_i - c_i^*) + M, \quad (3.18)$$

$$\|M\|_2 \leq f_1(\epsilon_1) \|c - c^*\|_2^2$$

and

$$\|J(c)^{-1}\|_2 \leq \|J^{*-1}\|_2 + f_2(\epsilon_1), \quad (3.19)$$

where $f_i(\epsilon_1) \geq 0$ ($i = 1, 2$) are continuous functions of ϵ_1 and $f_i(0) = 0$, $J(c) \in \mathbb{C}^{n \times n}$ is obtained by Step d) in Algorithm 3.2 at $c \in \mathbb{C}^n$.

On the other hand, we note

$$U(c)^H A(c) U(c) = H(c), \quad B^{(i)}(c)\pi^{(i)}(c) = Q^{(i)}(c)R^{(i)}(c),$$

where $B^{(i)}(c) = H(c) - \lambda_i^* I$. Therefore for $\tilde{c} \in \mathbb{C}^n$ close to c , we have

$$[U(c)^H A(\tilde{c}) U(c) - \lambda_i^* I] \pi^{(i)}(c) = Q^{(i)}(\tilde{c}) R^{(i)}(\tilde{c}),$$

$$R^{(i)}(\tilde{c}) = \begin{pmatrix} R_{11}^{(i)}(\tilde{c}) & R_{12}^{(i)}(\tilde{c}) \\ 0 & r_{nn}^{(i)}(\tilde{c}) \end{pmatrix}, \quad i = 1, \dots, n.$$

From Theorem 2.1, (3.18) and (3.19), it follows that there exist $\epsilon_2 > 0$ and $\epsilon_3 > 0$ such that if $\|c - c^*\|_2 < \epsilon_2$ then

$$\begin{aligned} r_{nn}^{(i)}(\tilde{c}) &= r_{nn}^{(i)}(c) + O(\|\tilde{c} - c\|_2) + f_3(\epsilon_2, \epsilon_3)\delta^{(i)}, \\ |\delta^{(i)}| &\leq \|\tilde{c} - c\|_2^2, \quad \text{if } \|\tilde{c} - c\|_2 \leq \epsilon_3, \end{aligned} \tag{3.20}$$

where $f_3(\epsilon_2, \epsilon_3) \geq 0$ is a continuous function of (ϵ_2, ϵ_3) and $f_3(0, 0) = 0$.

Now suppose $\|c^{(\nu)} - c^*\|_2 < \min\{\epsilon_2, \epsilon_3\}$, then from (3.20) we have (note $r_{nn}^{(i)}(c^*) = 0$)

$$\begin{aligned} 0 &= b^{(\nu)} + J^{(\nu)}(c^* - c^{(\nu)}) + f_4(\epsilon_2, \epsilon_3)\delta, \\ \|\delta\|_2 &\leq \|c^* - c^{(\nu)}\|_2^2, \end{aligned}$$

i.e.

$$c^* = c^{(\nu)} - J^{(\nu)-1}b^{(\nu)} - f_4(\epsilon_2, \epsilon_3)J^{(\nu)-1}\delta, \tag{3.21}$$

where $f_4(\epsilon_2, \epsilon_3) \geq 0$ is a continuous function of (ϵ_2, ϵ_3) and $f_4(0, 0) = 0$. (3.21) together with (3.19) and $c^{(\nu+1)} = c^{(\nu)} - J^{(\nu)-1}b^{(\nu)}$ lead to $\|c^{(\nu+1)} - c^*\|_2 = O(\|c^{(\nu)} - c^*\|_2^2)$. ■

§4. NUMERICAL EXAMPLES

Before giving our examples, it is necessary to present some perturbation results which can be used to measure the differences between the spectrum of λ^* and that of $A(c^{(s)})$, where $c^{(s)}$ is an accepted approximation to c^* .

By factorizations $(A(c^{(s)}) - \lambda_i^*I)\pi^{(i,s)} = Q^{(i,s)}R^{(i,s)}$, $i = 1, \dots, n$, we get

$$e_n^T Q^{(i,s)H} (A(c^{(s)}) - \lambda_i^*I) = r_{nn}^{(i,s)} e_n^T \pi^{(i,s)T}. \tag{4.1}$$

Therefore if $A(c)$ is Hermitian or normal then for any i ($1 \leq i \leq n$) there exists an eigenvalue λ of $A(c^{(s)})$ (see [5]) such that

$$|\lambda - \lambda_i^*| \leq |r_{nn}^{(i,s)}|. \tag{4.2}$$

Since $\lambda_i^* \neq \lambda_j^*$ ($i \neq j$), n intervals $|z - \lambda_i^*| \leq |r_{nn}^{(i,s)}|$ are disjointed from each other if $|r_{nn}^{(i,s)}|$ sufficiently small, thus n eigenvalues of $A(c^{(s)})$ are different from each other and satisfy

$$|\lambda_i - \lambda_i^*| \leq |r_{nn}^{(i,s)}|, i = 1, \dots, n. \tag{4.3}$$

For general matrices $A(c)$, just from (4.1) we are unable to give an estimation of differences between the spectrum of $A(c^{(s)})$ and λ^* , and therefore other information is needed. One of the compensating methods is that equations

$$(A(c^{(s)}) - \lambda_i^*I)p_i = r_i, \quad \|p_i\|_2 = 1, \quad 1 \leq i \leq n, \tag{4.4}$$

are also available, and approximately $\|r_i\|_2 \approx |r_{nn}^{(i,s)}|$. In fact this is possible since the smallest singular value of $A(c^{(s)}) - \lambda_i^*I$ is less than or equals $|r_{nn}^{(i,s)}|$, and thus a QRD with

column pivoting of $(A(c^{(s)}) - \lambda_i^* I)^H$ gives (4.4). Suppose now (4.1) and (4.4) are both available, then from [15] we know that there exist $E_i \in \mathbb{C}^{n \times n}$ ($i = 1, \dots, n$) with

$$\|E_i\|_2 = \max\{\|r_i\|_2, |r_{nn}^{(i,s)}|\}$$

such that λ_i^* is an eigenvalue of $A(c^{(s)}) - E_i$. If $|r_{nn}^{(i,s)}|$ sufficiently small (so are E_i), then the n eigenvalues of $A(c^{(s)}) - E_i$, $1 \leq i \leq n$ are different from each other since $\lambda_i^* \neq \lambda_j^*$ ($i \neq j$). Therefore from [5, p.176] we have that n eigenvalues λ_i ($i = 1, \dots, n$) of $A(c^{(s)})$ satisfy

$$\lambda_i = \lambda_i^* + O\left(\frac{\max\{\|r_i\|_2, |r_{nn}^{(i,s)}|\}}{|e_n^T Q^{(i,s)} p_i|}\right). \quad (4.5)$$

The following tests were conducted on IBM-PC/XT. Double precision arithmetic was used throughout. The starting points were chosen close to the solution, so that few iterations were required for convergence. A line search (or trust region strategy) would be essential to make the algorithms convergent in practical applications. However, we have not included these features and have concentrated on the local behavior of the algorithms.

Example 1([1]). This is an inverse additive problem. Here $n = 8$,

$$A_0 = \begin{pmatrix} 0 & 4 & -1 & 1 & 1 & 5 & -1 & 1 \\ 4 & 0 & -1 & 2 & 1 & 4 & -1 & 2 \\ -1 & -1 & 0 & 3 & 1 & 3 & -1 & 3 \\ 1 & 2 & 3 & 0 & 1 & 2 & -1 & 4 \\ 1 & 1 & 1 & 1 & 0 & 1 & -1 & 5 \\ 5 & 4 & 3 & 2 & 1 & 0 & -1 & 6 \\ -1 & -1 & -1 & -1 & -1 & -1 & 0 & 7 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 0 \end{pmatrix}, \quad A_k = e_k e_k^T, k = 1, \dots, 8,$$

$$A(c) = A_0 + \sum_{i=1}^8 c_i A_i,$$

$$\lambda^* = (10, 20, 30, 40, 50, 60, 70, 80),$$

$$c^* = (11.907876, 19.705522, 30.545498, 40.062657,$$

$$51.587140, 64.702131, 70.170676, 71.318499).$$

With the starting point $c^{(0)} = (10, 20, 30, 40, 50, 60, 70, 80)$, Algorithm 3.2 diverges, while Algorithm 3.1 converges. The computational results of Algorithm 3.1 are displayed in Table 4.1, where $\delta c^{(\nu)} = (\delta c_1^{(\nu)}, \dots, \delta c_n^{(\nu)})^T \in \mathbb{C}^{n \times n}$ is defined by $c^{(\nu+1)} = c^{(\nu)} + \delta c^{(\nu)}$, $\lambda_1^{(\nu)} \leq \dots \leq \lambda_n^{(\nu)}$ are n eigenvalues of $A(c^{(\nu)})$.

Table 4.1

No. (ν)	Algorithm 3.1		Method I in [1]
	$\max\{ r_{nn}^{(i,\nu)} \}$	$\max\{ \delta c_i^{(\nu)} \}$	$\max\{ \lambda_i^{(\nu)} - \lambda_i^* \}$
0	6.4E + 00	8.5E + 00	6.4E + 00
1	7.1E - 01	1.2E + 00	8.9E - 01
2	3.9E - 02	1.0E - 01	1.0E - 01
3	4.4E - 04	1.0E - 03	2.7E - 03
4	4.7E - 08	1.1E - 07	2.3E - 06
5	6.5E - 15	8.2E - 15	1.7E - 12

With the starting point $c^{(0)} = (10, 20, 30, 40, 50, 60, 70, 79)$, Algorithms 3.1 and 3.2 are both converge, and their computational results are displayed in Table 4.2. Here the column corresponding to δ_ν displays approximately the maximum of modulus among southeast elements of n upper triangular matrices obtained by computing QRDs with column pivoting of $A(c^{(\nu)}) - \lambda_i^* I$, $i = 1, \dots, 8$ in the process of computing with Algorithm 3.2, in order that we can compare our two algorithms. We see that Algorithm 3.1 converges more rapidly than Algorithm 3.2 in this test. It seems to me that such observation is valid all the time.

Table 4.2

No. (ν)	Algorithm 3.1		Algorithm 3.2		
	$\max\{ r_{nn}^{(i,\nu)} \}$	$\max\{ \delta c_i^{(\nu)} \}$	$\max\{ r_{nn}^{(i,\nu)} \}$	$\max\{ \delta c_i^{(\nu)} \}$	δ_ν
0	5.58E + 00	7.50E + 00	1.32E + 01	1.29E + 01	5.58E + 00
1	6.28E - 01	8.64E - 01	4.48E + 00	1.01E + 01	3.47E + 00
2	3.67E - 02	8.59E - 02	9.64E + 00	5.50E + 00	3.36E + 00
3	3.59E - 04	8.22E - 04	2.81E + 00	6.82E - 01	4.13E - 01
4	3.13E - 08	7.32E - 08	4.56E + 00	6.88E - 02	4.88E - 02
5	7.34E - 15	8.48E - 15	4.08E - 03	7.46E - 05	5.11E - 05
6			1.74E - 07	2.53E - 09	1.87E - 09
7			1.39E - 12	2.01E - 14	1.63E - 12

Example 2([4]). $n = 5$,

$$A_0 = \begin{pmatrix} 2 & -0.08 & & & \\ -0.03 & 2 & -0.08 & & \\ & -0.03 & 2 & -0.08 & \\ & & -0.03 & 2 & -0.08 \\ & & & -0.03 & 2 \end{pmatrix}, \quad A_k = r_k e_k^T, k = 1, \dots, 5$$

with $r_k \in \mathbb{R}^5$ defined by

$$R = \sum_{i=1}^5 r_i e_i^T = \begin{pmatrix} 1 & 0 & 0.01 & -0.02 & 0.03 \\ -0.03 & 1 & 0 & 0.01 & -0.02 \\ 0.02 & -0.03 & 1 & 0 & 0.01 \\ -0.01 & 0.02 & -0.03 & 1 & 0 \\ 1 & -0.01 & 0.02 & -0.03 & 1 \end{pmatrix}$$

and

$$A(c) = A_0 + \sum_{i=1}^5 c_i A_i \quad ,$$

$$\lambda^* = (0 + \delta, 1 - \delta, 2 + \delta, 3 - \delta, 4).$$

Table 4.3 displays the computational results of Algorithms 3.1 and 3.2 with the starting point $c^{(0)} = (2, 1, 0, -1, -2)$ and $\delta = 0$.

Table 4.3

No. (ν)	Algorithm 3.1		Algorithm 3.2		
	$\max\{ r_{nn}^{(i,\nu)} \}$	$\max\{ \delta c_i^{(\nu)} \}$	$\max\{ r_{nn}^{(i,\nu)} \}$	$\max\{ \delta c_i^{(\nu)} \}$	δ_ν
0	$7.15E - 03$	$7.18E - 03$	$4.30E - 02$	$7.18E - 03$	$7.15E - 03$
1	$3.768 - 07$	$3.71E - 07$	$1.82E - 03$	$8.89E - 06$	$8.87E - 06$
2	$1.36E - 15$	$1.37E - 15$	$4.55E - 08$	$7.97E - 11$	$7.96E - 11$
$c^* = (1.99282, 1.0028, 0.00236, -0.99788, -2.00012)$					

with the same starting point but $\delta = 0.441$, however Algorithm 3.2 fails to converge. Table 4.4 displays the computational results of Algorithm 3.1.

Table 4.4: Algorithm 3.1

No. (ν)	$\max\{ r_{nn}^{(i,\nu)} \}$	$\max\{ \delta c_i^{(\nu)} \}$
0	$4.4E - 01$	$4.4E - 01$
1	$2.2E - 02$	$3.3E - 02$
2	$3.9E - 03$	$1.0E - 02$
3	$6.8E - 04$	$3.1E - 03$
4	$1.1E - 04$	$5.5E - 04$
5	$3.4E - 06$	$1.9E - 05$
6	$4.1E - 09$	$2.4E - 08$
7	$5.8E - 15$	$3.6E - 14$
$c^* = (1.99510, 0.511492, 0.49191, -1.43089, -1.56761)$		

§5. CONCLUDING REMARKS

(i) Throughout the paper, we assume that the studied problems appear in the complex field. We can find out easily that the complex field can be replaced by the real field, i.e., all \mathbb{C}, \mathbb{C}^n , etc. can be replaced by \mathbb{R}, \mathbb{R}^n , etc.

(ii) Our algorithms are derived for general matrices, while algorithms in [1] can only be applied to inverse problems for real symmetric matrices or Hermitian matrices. (Yet, they can be generalized to inverse problems for general matrices in a straightforward way.)

ACKNOWLEDGEMENT. *The author would like to thank Professor Sun Ji-guang for his supervision and guidance.*

REFERENCES

- [1] Friedland, S., Nocedal, J. and Overton, M.L., The formulation and analysis of numerical methods for inverse eigenvalue problems, *SIAM J. Numer. Anal.*, **24**(1987), 634-667.
- [2] Friedland, S., inverse eigenvalue problems, *Linear Algebra Appl.*, **17**(1977), 15-51.
- [3] Friedland, S., On multiplicative inverse eigenvalue problems for matrices, *Linear Algebra Appl.*, **12**(1975), 127-137.
- [4] Biegler-König, F.W., A newton iteration process for inverse eigenvalue problems, *Numer. Math.*, **37**(1981), 349-354.
- [5] Sun, Ji-guang, *Matrix Perturbation Analysis*, Science Press, Beijing, 1987 (Chinese).
- [6] Sun, Ji-guang, On algebraic inverse eigenvalue problems, Computing Center, Academia Sinica, 1985.
- [7] Sun, Ji-guang, On the sufficient conditions for the solubility of algebraic inverse eigenvalue problems, *Math. Numer. Sinica*, **9**:1 (1987), 51-59 (Chinese).
- [8] Ye Qiang, A class of iterative algorithms for solving inverse eigenvalue problems, *Math. Numer. Sinica*, **9**:2 (1987), 143-153 (Chinese).
- [9] Golub, G.H. and van Loan, C.F., *Matrix Computations*, the John Hopkins Univ. Press, 1983.
- [10] Nocedal, J. and Overton, M.L., Numerical methods for solving inverse eigenvalue problems, Lect. Notes in Math., No.1005, V.Pereyra and A.Reinoza, eds. Springer-Verlag, New York, Berlin, 1983, 212-226.
- [11] Downing, A.C., Housholder, A.S., Some inverse characteristic value problems, *J. Assoc. Comput. Math.*, **3**(1956), 203-207.
- [12] Li, Ren-cang, QR decomposition and nonlinear eigenvalue problems, *Math. Numer. Sinica*, **11**:4 (1989), 374-385 (Chinese).
- [13] Kublanovskaya, V.N., On an approach to the solution of the generalized latent value problems for λ -matrices, *SIAM J. Numer. Anal.*, **7**(1970), 532-537.
- [14] Jain, N.K. and Singhal, K., On Kublanovskaya's approach to the solution of the generalized latent value problems for functional λ -matrices, *SIAM J. Numer. Anal.*, **20**(1983), 1062-1067.
- [15] Kahan, W.M., Parlett, B.N. and Jiang, Erxiong, Residual bounds on approximate eigen-systems of nonnormal matrices, *SIAM J. Numer. Anal.*, **19**(1982), 470-484.