



A Ulm-like method for inverse eigenvalue problems [☆]

W.P. Shen ^{a,*}, C. Li ^{b,c}, X.Q. Jin ^d

^a Department of Mathematics, Zhejiang Normal University, Jinhua 321004, PR China

^b Department of Mathematics, Zhejiang University, Hangzhou 310027, PR China

^c Department of Mathematics, College of Sciences, King Saud University, P.O. Box 2455, Riyadh 11451, Saudi Arabia

^d Department of Mathematics, University of Macau, Macao, PR China

ARTICLE INFO

Article history:

Received 17 March 2010

Received in revised form 14 August 2010

Accepted 2 November 2010

Available online 4 November 2010

Keywords:

Nonlinear equation

Inverse eigenvalue problem

Inexact Newton-like method

Ulm-like method

ABSTRACT

We propose a Ulm-like method for solving inverse eigenvalue problems, which avoids solving approximate Jacobian equations comparing with other known methods. A convergence analysis of this method is provided and the R-quadratic convergence property is proved under the assumption of the distinction of given eigenvalues. Numerical experiments as well as the comparison with the inexact Newton-like method are given in the last section.

© 2010 IMACS. Published by Elsevier B.V. All rights reserved.

1. Introduction

Over the years there has been considerable interest in deriving and analyzing the theory or algorithms for the inverse eigenvalue problem (IEP) which is defined as follows. Let $\mathbf{c} = (c_1, c_2, \dots, c_n)^T \in \mathbb{R}^n$ and $\{A_i\}_{i=1}^n$ be n real symmetric $n \times n$ matrices. Define

$$A(\mathbf{c}) := \sum_{i=1}^n c_i A_i \quad (1.1)$$

and denote its eigenvalues by $\{\lambda_i(\mathbf{c})\}_{i=1}^n$ with the order $\lambda_1(\mathbf{c}) \leq \lambda_2(\mathbf{c}) \leq \dots \leq \lambda_n(\mathbf{c})$. Let $\{\lambda_i^*\}_{i=1}^n$ be given with $\lambda_1^* \leq \lambda_2^* \leq \dots \leq \lambda_n^*$. Then the IEP considered here is to find a vector $\mathbf{c}^* \in \mathbb{R}^n$ such that

$$\lambda_i(\mathbf{c}^*) = \lambda_i^* \quad \text{for each } i = 1, 2, \dots, n. \quad (1.2)$$

The vector \mathbf{c}^* is called a solution of IEP (1.2).

This type of inverse problems arises in a variety of applications such as inverse Sturm–Liouville's problem, inverse vibrating string problem, nuclear spectroscopy and molecular spectroscopy (see [3,4,8,10,15,16,19,21,23,25–27,30,31]). In particular, a recent survey paper on structured inverse eigenvalue problems by Chu and Golub (see [8]) is a good reference for these applications. In many of these applications, the problem size n can be large, for example, large inverse Toeplitz eigenvalue problems and large discrete inverse Sturm–Liouville's problems considered in [27,8].

[☆] This project was supported in part by the National (grant 10731060) and Zhejiang Provincial (grant Y6100375) Natural Science Foundation of China and by the grant UL020/08-Y2/MAT/JXQ01/FST from University of Macau.

* Corresponding author.

E-mail addresses: shenweiping@zjnu.cn (W.P. Shen), cli@zju.edu.cn (C. Li), xqjin@umac.mo (X.Q. Jin).

Recall that solving the IEP (1.2) is equivalent to solving the equation $\mathbf{f}(\mathbf{c}) = \mathbf{0}$ on \mathbb{R}^n , where the function $\mathbf{f}: \mathbb{R}^n \rightarrow \mathbb{R}^n$ is defined by

$$\mathbf{f}(\mathbf{c}) := (\lambda_1(\mathbf{c}) - \lambda_1^*, \lambda_2(\mathbf{c}) - \lambda_2^*, \dots, \lambda_n(\mathbf{c}) - \lambda_n^*)^T \quad \text{for any } \mathbf{c} \in \mathbb{R}^n. \tag{1.3}$$

Based on this equivalence, Newton’s method can be applied to solving the IEP, and it converges quadratically (see [13,22,33]). As it is well known, each iteration of Newton’s method involves solving a complete eigenproblem for the matrix $A(\mathbf{c})$. To overcome this drawback, different Newton-like methods, where each outer iteration, instead of the exact eigenvectors of the matrix $A(\mathbf{c})$, adopts approximations to them, have been proposed and studied in [6,7,13,33]. In particular, Friedland, Nocedal, and Overton considered in [13] a type of Newton-like method where the approximate eigenvectors were found by using the one-step inverse power method. The quadratic convergence rate of this type of Newton-like method was re-proved in paper [6]. To alleviate the over-solving problem, Chan, Chung, and Xu proposed in [5] an inexact version of this Newton-like method considered in [13]. Their method stops the inner iterations before convergence. Suitable inner tolerances for systems were set in that paper and a superlinear convergence rate was obtained.

The purpose of the present paper is, motivated by Moser’s method and Ulm’s method (see [11,14,18,20,24,28,29,32,34]), to propose a Ulm-like method for solving the IEP, which avoids solving the Jacobian equations in each outer iteration. Under the classical assumption (which is also used in [5]) that the given eigenvalues are distinct and the Jacobian matrix $J(\mathbf{c}^*)$ is invertible, we prove that this method converges with R-quadratic convergence. Comparing with the inexact Newton-like method in [5], the Ulm-like method seems more stable, and reduces the difficulty though they have the same costs. Numerical experiments are given in the last section to illustrate the comparison with the inexact Newton-like method. In particular, an example for which the Ulm-like method converges but not the inexact Newton-like method is provided.

2. Ulm-like method

Let $[1, n]$ denote the set of $\{1, 2, \dots, n\}$. As usual, let $\mathbb{R}^{n \times n}$ denote the set of all real $n \times n$ matrices. Let $\|\cdot\|$ denote the 2-norm in \mathbb{R}^n . The induced 2-norm in $\mathbb{R}^{n \times n}$ is also denoted by $\|\cdot\|$, i.e.,

$$\|A\| := \sup_{\mathbf{x} \in \mathbb{R}^n, \mathbf{x} \neq \mathbf{0}} \frac{\|A\mathbf{x}\|}{\|\mathbf{x}\|} \quad \text{for each } A \in \mathbb{R}^{n \times n}.$$

Let $\|\cdot\|_F$ denote the Frobenius norm in $\mathbb{R}^{n \times n}$. Then

$$\|A\| \leq \|A\|_F \quad \text{for each } A \in \mathbb{R}^{n \times n}. \tag{2.1}$$

Let $\mathbf{c} = (c_1, c_2, \dots, c_n)^T \in \mathbb{R}^n$ and $\{A_i\}_{i=1}^n \subset \mathbb{R}^{n \times n}$ be symmetric. As in (1.1), define

$$A(\mathbf{c}) = \sum_{i=1}^n c_i A_i.$$

Let $\lambda_1(\mathbf{c}) \leq \lambda_2(\mathbf{c}) \leq \dots \leq \lambda_n(\mathbf{c})$ be the eigenvalues of the matrix $A(\mathbf{c})$, and let $\{\mathbf{q}_i(\mathbf{c})\}_{i=1}^n$ be the normalized eigenvectors corresponding to $\{\lambda_i(\mathbf{c})\}_{i=1}^n$. Define $J(\mathbf{c}) = ([J(\mathbf{c})]_{ij})$ by

$$[J(\mathbf{c})]_{ij} := \mathbf{q}_i(\mathbf{c})^T A_j \mathbf{q}_i(\mathbf{c}) \quad \text{for any } i, j \in [1, n]. \tag{2.2}$$

Let $\{\lambda_i^*\}_{i=1}^n$ be given with $\lambda_1^* \leq \lambda_2^* \leq \dots \leq \lambda_n^*$ and write $\boldsymbol{\lambda}^* = (\lambda_1^*, \lambda_2^*, \dots, \lambda_n^*)^T$. Let \mathbf{c}^* be a solution of the IEP, i.e.,

$$\lambda_i(\mathbf{c}^*) = \lambda_i^* \quad \text{for each } i \in [1, n].$$

As shown in [5,13], in the case when the given eigenvalues $\{\lambda_i^*\}_{i=1}^n$ are distinct, the eigenvalues of $A(\mathbf{c})$ are distinct too for any point \mathbf{c} in some neighborhood of \mathbf{c}^* . It follows that the function $\mathbf{f}(\cdot)$ is analytic in the same neighborhood, and $J(\mathbf{c})$ is the Jacobian matrix of \mathbf{f} at \mathbf{c} in this neighborhood (see [5]). Recall that Newton’s method, which converges quadratically (see [22,33]), involves solving a complete eigenproblem for the matrix $A(\mathbf{c})$. However, if we only compute it approximately, we may still have fast convergence. This results in the following Newton-like method, which was proposed in [13].

Algorithm 1 (Newton-like method).

1. Given \mathbf{c}^0 , compute the eigen-decomposition of $A(\mathbf{c}^0)$ to get $\mathbf{p}_i^0 = \mathbf{q}_i(\mathbf{c}^0)$ for each $i \in [1, n]$ and solve \mathbf{c}^1 from the Jacobian equation

$$J(\mathbf{c}^0)\mathbf{c}^1 = \boldsymbol{\lambda}^*.$$

2. For $k = 1, 2, \dots$ until convergence, do:
 - (a) Compute \mathbf{v}_i^k in the one-step inverse power method:

$$(A(\mathbf{c}^k) - \lambda_i^* I)\mathbf{v}_i^k = \mathbf{p}_i^{k-1} \quad \text{for each } i \in [1, n]. \tag{2.3}$$

(b) Normalize \mathbf{v}_i^k to obtain an approximate eigenvector \mathbf{p}_i^k of $A(\mathbf{c}^k)$:

$$\mathbf{p}_i^k := \frac{\mathbf{v}_i^k}{\|\mathbf{v}_i^k\|} \quad \text{for each } i \in [1, n]. \tag{2.4}$$

(c) Form the approximate Jacobian matrix J_k :

$$[J_k]_{ij} := (\mathbf{p}_i^k)^T A_j \mathbf{p}_i^k \quad \text{for each } i, j \in [1, n]. \tag{2.5}$$

(d) Compute \mathbf{c}^{k+1} from the approximate Jacobian equation:

$$J_k \mathbf{c}^{k+1} = \lambda^*. \tag{2.6}$$

In [6,13], it was proved that the Newton-like method Algorithm 1 converges quadratically. Note that, in Algorithm 1, systems (2.3) and (2.6) are solved exactly. Usually, one solves these systems by iterative methods, in particular in the case when n is large. While iterative methods may bring an over-solving problem in the sense that the last few iterations before convergence are usually insignificant as far as the convergence of the outer iteration is concerned. This over-solving of the inner iterations will cause a waste of time and does not improve the efficiency of the whole method. To alleviate the over-solving problem and improve the efficiency in solving the IEP, systems (2.3) and (2.6) were solved in [5] approximately rather than exactly, and the following inexact Newton-like method was proposed there.

Algorithm 2 (Inexact Newton-like method).

1. Same as 1 in Algorithm 1.
2. For $k = 1, 2, \dots$ until convergence, do:
 - (a) Solve \mathbf{v}_i^k inexactly in the one-step inverse power method:

$$(A(\mathbf{c}^k) - \lambda_i^* I) \mathbf{v}_i^k = \mathbf{p}_i^{k-1} + \mathbf{t}_i^k \quad \text{for each } i \in [1, n], \tag{2.7}$$

until the residual \mathbf{t}_i^k satisfies

$$\|\mathbf{t}_i^k\| \leq \frac{1}{4} \quad \text{for each } i \in [1, n]. \tag{2.8}$$

- (b) Same as (b) in Algorithm 1.
- (c) Same as (c) in Algorithm 1.
- (d) Solve \mathbf{c}^{k+1} inexactly from the approximate Jacobian equation:

$$J_k \mathbf{c}^{k+1} = \lambda^* + \mathbf{r}_k \tag{2.9}$$

until the residual \mathbf{r}_k satisfies

$$\|\mathbf{r}_k\| \leq \left(\max_{1 \leq i \leq n} \frac{1}{\|\mathbf{v}_i^k\|} \right)^\beta \quad \text{for } 1 < \beta \leq 2.$$

Under the assumption that the given eigenvalues $\{\lambda_i^*\}_{i=1}^n$ are distinct and the Jacobian matrix $J(\mathbf{c}^*)$ is invertible, it was proved in [5] that the inexact Newton-like method converges locally with a convergence rate β . In each outer iteration of the inexact Newton-like method, an approximate Jacobian equation, the tolerance for which is of $O(\|\mathbf{c}^k - \mathbf{c}^*\|^\beta)$, is required to solve. This still can be costly sometimes especially when \mathbf{c}^k is close to the solution \mathbf{c}^* . Furthermore, solving system (2.9) may involve some instability problem or preconditioning problem. In fact, a numerical example has been provided in [2] to show that the approximate Jacobian equation (2.9) is increasingly ill-conditioned and the inexact Newton-like method fails to converge.

Moser’s method (see [18,24,28]) to solve operator equations in Banach spaces is defined as follows. Let X, Y be (real or complex) Banach spaces, and let $D \subseteq X$ be an open subset. Consider the general operator equation:

$$f(x) = 0, \tag{2.10}$$

where $f : D \subseteq X \rightarrow Y$ is a nonlinear operator with continuous Fréchet derivative f' . Given $x_0 \in D$ and $B_0 \in \mathcal{L}(Y, X)$, Moser’s method to find solutions of Eq. (2.10) is defined as follows:

$$\begin{cases} x_{k+1} = x_k - B_k f(x_k) \\ B_{k+1} = 2B_k - B_k f'(x_k) B_k \end{cases} \quad \text{for each } k = 0, 1, \dots$$

The convergence rate of Moser’s method is $(1 + \sqrt{5})/2 = 1.61 \dots$ (see [24]). However, quadratic convergence rate can be obtained when the sequence $\{B_k\}$ is generated by

$$B_{k+1} = 2B_k - B_k f'(x_{k+1}) B_k \quad \text{for each } k = 0, 1, \dots$$

This is Ulm’s method introduced in [32] and has been further studied in [11,14,18,20,28,29,34]. R-quadratic convergence of Ulm’s method was established in [11,20,34] under the classical assumption that the derivative f' is Lipschitz continuous around the solution. Compared with Newton’s method, the advantage of Moser’s method and Ulm’s method is that Jacobian equations are not required to solve in each step. Motivated by Ulm’s method, we propose the following Ulm-like method for solving the IEP, which also avoids solving the approximate Jacobian equations in each step.

Algorithm 3 (Ulm-like method).

1. Let $\mathbf{c}^0 \in \mathbb{R}^n$ and $B_0 \in \mathbb{R}^{n \times n}$ be such that

$$\|I - B_0 J(\mathbf{c}^0)\| \leq \mu, \tag{2.11}$$

where μ is a positive constant. Compute the eigen-decomposition of $A(\mathbf{c}^0)$ to get

$$\mathbf{p}_i^0 := \mathbf{q}_i(\mathbf{c}^0) \quad \text{for each } i \in [1, n] \tag{2.12}$$

and define \mathbf{c}^1 by

$$\mathbf{c}^1 := \mathbf{c}^0 - B_0(J(\mathbf{c}^0)\mathbf{c}^0 - \lambda^*). \tag{2.13}$$

2. For $k = 1, 2 \dots$ until convergence, do:
 - (a) Same as (a) in Algorithm 2.
 - (b) Same as (b) in Algorithm 1.
 - (c) Same as (c) in Algorithm 1.
 - (d) Define the matrix B_k and \mathbf{c}^{k+1} respectively by

$$B_k := 2B_{k-1} - B_{k-1} J_k B_{k-1}$$

and

$$\mathbf{c}^{k+1} := \mathbf{c}^k - B_k(J_k \mathbf{c}^k - \lambda^*). \tag{2.14}$$

Remark 2.1. The main difference of the Ulm-like method and the inexact Newton-like method is that the step of solving the Jacobian equation (2.9) in the inexact Newton-like method is replaced by computing the product of matrices, the operation cost of which is still $O(n^3)$, the same as that of solving the Jacobian equation. However, computing the product of matrices is simpler than solving equations and has no instability problem caused by ill-conditioning in solving equations (cf. [2], where an example was presented to show that $\{J_k\}$ are increasingly ill-conditioned even in the case when the Ulm-like method converges). Example 4.2 below in Section 4 provides some cases when the Ulm-like method converges but not the inexact Newton-like method even if the condition numbers of $\{J_k\}$ are not very large. Therefore, the Ulm-like method reduces significantly the difficulty of the problem. In particular, the parallel computation techniques can be applied in the Ulm-like method to improve the computational efficiency.

3. Convergence analysis

In this section, we carry on a convergence analysis of the Ulm-like method. Let $\{\lambda_i^*\}_{i=1}^n$ be given with $\lambda_1^* \leq \lambda_2^* \leq \dots \leq \lambda_n^*$, and let \mathbf{c}^* be a solution of the IEP. As the standard assumption in [5], we assume that the given eigenvalues $\{\lambda_i^*\}_{i=1}^n$ are distinct and the Jacobian matrix $J(\mathbf{c}^*)$ is invertible. Let \mathbf{c}^k be the k th iteration of the method, and let $\{\lambda_i(\mathbf{c}^k)\}_{i=1}^n$ and $\{\mathbf{q}_i(\mathbf{c}^k)\}_{i=1}^n$ be the eigenvalues and normalized eigenvectors of $A(\mathbf{c}^k)$ respectively, i.e.,

$$A(\mathbf{c}^k)\mathbf{q}_i(\mathbf{c}^k) = \lambda_i(\mathbf{c}^k)\mathbf{q}_i(\mathbf{c}^k) \quad \text{and} \quad \mathbf{q}_i(\mathbf{c}^k)^T \mathbf{q}_j(\mathbf{c}^k) = \begin{cases} 1, & \text{if } i = j; \\ 0, & \text{if } i \neq j. \end{cases}$$

Below, we prove that if B_0 approximates $J(\mathbf{c}^0)^{-1}$ and the initial guess \mathbf{c}^0 is closed to the solution \mathbf{c}^* , then the sequence $\{\mathbf{c}^k\}$ generated by the Ulm-like method converges locally to \mathbf{c}^* with R-convergence rate 2. For this purpose, we need the following three lemmas. The first two lemmas were presented in [5].

Lemma 3.1. Suppose that $\{\lambda_i^*\}_{i=1}^n$ are distinct. Then there exist positive numbers δ_0, ρ_0 , and γ such that the following assertions hold for each $\mathbf{c} \in \mathbf{B}(\mathbf{c}^*, \delta_0)$.

- (i) $|\lambda_i(\mathbf{c}) - \lambda_i^*| \leq \rho_0 \|\mathbf{c} - \mathbf{c}^*\|$ for each $i \in [1, n]$.
- (ii) $\|\mathbf{q}_i(\mathbf{c}) - \mathbf{q}_i(\mathbf{c}^*)\| \leq \rho_0 \|\mathbf{c} - \mathbf{c}^*\|$ for each $i \in [1, n]$.
- (iii) $|\lambda_i(\mathbf{c}) - \lambda_j^*| \geq \gamma$ for any $i, j \in [1, n]$ and $i \neq j$.

Lemma 3.2. Let $\{\boldsymbol{\omega}_i\}_{i=1}^n \subset \mathbb{R}^n$ be unit vectors approximating $\mathbf{q}_i(\mathbf{c}^*)$, and let J_ω be the matrix defined by $[J_\omega]_{ij} = (\boldsymbol{\omega}_i)^T A_j \boldsymbol{\omega}_i$ for $i, j \in [1, n]$. Then

$$\|J_\omega \mathbf{c}^* - J(\mathbf{c}^*) \mathbf{c}^*\| \leq 2n \cdot \max_i |\lambda_i^*| \cdot \max_i \|\boldsymbol{\omega}_i - \mathbf{q}_i(\mathbf{c}^*)\|^2.$$

For the following key lemma, we define

$$J_0 := J(\mathbf{c}^0), \quad \mathbf{p}_i^{-1} := \mathbf{q}_i(\mathbf{c}^0) \quad \text{for each } i \in [1, n] \tag{3.1}$$

and

$$\epsilon_i^k := \text{sign}(\mathbf{q}_i(\mathbf{c}^k)^T \mathbf{p}_i^k) \quad \text{for any } i \in [1, n], \text{ and } k = 0, 1, \dots \tag{3.2}$$

Lemma 3.3. Let $\delta_0 \in (0, 1)$ and $\rho_0 \in (0, +\infty)$ be determined by Lemma 3.1. Let δ_1 be such that

$$0 < \delta_1 < \min \left\{ \delta_0, \frac{1}{H_1 \|J(\mathbf{c}^*)^{-1}\|} \right\}, \tag{3.3}$$

where

$$H_1 := 2n\rho_0(1 + 8/\gamma) \cdot \max_j \|A_j\|. \tag{3.4}$$

Let $k = 0, 1, \dots$, be fixed. Suppose that

$$\|\mathbf{c}^k - \mathbf{c}^*\| \leq \delta_1 \tag{3.5}$$

and

$$|\mathbf{q}_i(\mathbf{c}^k)^T \mathbf{p}_i^{k-1}| \geq \frac{1}{2} \quad \text{for each } i \in [1, n]. \tag{3.6}$$

Then the following assertions hold.

- (i) $\|\epsilon_i^k \mathbf{p}_i^k - \mathbf{q}_i(\mathbf{c}^k)\| \leq \frac{8\rho_0}{\gamma} \|\mathbf{c}^k - \mathbf{c}^*\|$ for each $i \in [1, n]$.
- (ii) $\|\epsilon_i^k \mathbf{p}_i^k - \mathbf{q}_i(\mathbf{c}^*)\| \leq (1 + \frac{8}{\gamma})\rho_0 \|\mathbf{c}^k - \mathbf{c}^*\|$ for each $i \in [1, n]$.
- (iii) $\|J_k - J(\mathbf{c}^*)\| \leq H_1 \|\mathbf{c}^k - \mathbf{c}^*\|$.
- (iv) $\|J_k^{-1}\| \leq \frac{\|J(\mathbf{c}^*)^{-1}\|}{1 - H_1 \|J(\mathbf{c}^*)^{-1}\| \delta_1}$.

Proof. Let $i, j \in [1, n]$. Then, by definitions of $[J_k]_{ij}$ and $[J(\mathbf{c}^*)]_{ij}$ (cf. (2.5) and (2.2)), we have

$$\begin{aligned} |[J_k]_{ij} - [J(\mathbf{c}^*)]_{ij}| &= |(\epsilon_i^k \mathbf{p}_i^k - \mathbf{q}_i(\mathbf{c}^*))^T A_j \epsilon_i^k \mathbf{p}_i^k - \mathbf{q}_i(\mathbf{c}^*)^T A_j (\mathbf{q}_i(\mathbf{c}^*) - \epsilon_i^k \mathbf{p}_i^k)| \\ &\leq 2\|A_j\| \cdot \|\epsilon_i^k \mathbf{p}_i^k - \mathbf{q}_i(\mathbf{c}^*)\|. \end{aligned}$$

Hence

$$\|J_k - J(\mathbf{c}^*)\| \leq \|J_k - J(\mathbf{c}^*)\|_F \leq 2n \cdot \max_j \|A_j\| \cdot \|\epsilon_i^k \mathbf{p}_i^k - \mathbf{q}_i(\mathbf{c}^*)\|. \tag{3.7}$$

Thus, if the assertion (ii) holds, then the assertion (iii) holds by the definition of H_1 , and the following inequality holds:

$$\|J(\mathbf{c}^*)^{-1}\| \cdot \|J_k - J(\mathbf{c}^*)\| \leq H_1 \|J(\mathbf{c}^*)^{-1}\| \delta_1 < 1.$$

This together with Banach's Lemma implies that the assertion (iv) holds too. Therefore, to complete the proof, it suffices to prove assertions (i) and (ii).

By assumptions (3.3) and (3.5), it is easy to see that Lemma 3.1 is applicable for \mathbf{c}^k in place of \mathbf{c} and so assertions (i)–(iii) in Lemma 3.1 hold for $\mathbf{c} = \mathbf{c}^k$. Note that $\mathbf{p}_i^0 = \mathbf{q}_i(\mathbf{c}^0)$ and $\epsilon_i^0 = \text{sign}(\mathbf{q}_i(\mathbf{c}^0)^T \mathbf{p}_i^0) = 1$ for each $i \in [1, n]$ (cf. (2.12) and (3.2)). Thus, for $k = 0$, the assertion (ii) holds by Lemma 3.1(ii) while the assertion (i) is trivial.

Below we consider the case of $k > 0$. Since $\{\mathbf{q}_j(\mathbf{c}^k)\}_{j=1}^n$ is an orthonormal basis, there exist n numbers $\{\xi_{ij}^k\}_{j=1}^n \subset \mathbb{R}$ such that

$$\mathbf{p}_i^{k-1} + \mathbf{t}_i^k = \sum_{j=1}^n \xi_{ij}^k \mathbf{q}_j(\mathbf{c}^k). \tag{3.8}$$

Hence

$$\sum_{j=1}^n (\xi_{ij}^k)^2 = \|\mathbf{p}_i^{k-1} + \mathbf{t}_i^k\|^2 \leq (\|\mathbf{p}_i^{k-1}\| + \|\mathbf{t}_i^k\|)^2 \leq (1 + 1/4)^2 \leq 2 \tag{3.9}$$

and

$$|\xi_{ii}^k| = |\mathbf{q}_i(\mathbf{c}^k)^T (\mathbf{p}_i^{k-1} + \mathbf{t}_i^k)| \geq |\mathbf{q}_i(\mathbf{c}^k)^T \mathbf{p}_i^{k-1}| - |\mathbf{q}_i(\mathbf{c}^k)^T \mathbf{t}_i^k| \geq \frac{1}{2} - \frac{1}{4} = \frac{1}{4} \tag{3.10}$$

as $|\mathbf{q}_i(\mathbf{c}^k)^T \mathbf{t}_i^k| \leq \|\mathbf{t}_i^k\| \leq \frac{1}{4}$ by (2.8) and $|\mathbf{q}_i(\mathbf{c}^k)^T \mathbf{p}_i^{k-1}| > \frac{1}{2}$ by (3.6). Note by (2.7) that

$$\mathbf{v}_i^k = (A(\mathbf{c}^k) - \lambda_i^* I)^{-1} (\mathbf{p}_i^{k-1} + \mathbf{t}_i^k)$$

and note also that $\mathbf{q}_j(\mathbf{c}^k)$ is an eigenvector of $(A(\mathbf{c}^k) - \lambda_i^* I)^{-1}$ corresponding to $(\lambda_j(\mathbf{c}^k) - \lambda_i^*)^{-1}$. It follows from (3.8) that

$$\mathbf{v}_i^k = \sum_{j=1}^n \xi_{ij}^k (A(\mathbf{c}^k) - \lambda_i^* I)^{-1} \mathbf{q}_j(\mathbf{c}^k) = \sum_{j=1}^n \frac{\xi_{ij}^k}{\lambda_j(\mathbf{c}^k) - \lambda_i^*} \mathbf{q}_j(\mathbf{c}^k).$$

Consequently,

$$\|\mathbf{v}_i^k\| = \frac{|\xi_{ii}^k|}{|\lambda_i(\mathbf{c}^k) - \lambda_i^*|} \left[1 + \sum_{j \neq i} \frac{(\xi_{ij}^k)^2 (\lambda_i(\mathbf{c}^k) - \lambda_i^*)^2}{(\xi_{ij}^k)^2 (\lambda_j(\mathbf{c}^k) - \lambda_i^*)^2} \right]^{\frac{1}{2}}$$

and

$$\mathbf{q}_i(\mathbf{c}^k)^T \mathbf{v}_i^k = \frac{\xi_{ii}^k}{\lambda_i(\mathbf{c}^k) - \lambda_i^*}.$$

Thus, by the definitions of \mathbf{p}_i^k and ϵ_i^k in (2.4) and (3.2) respectively,

$$0 \leq \epsilon_i^k \mathbf{q}_i(\mathbf{c}^k)^T \mathbf{p}_i^k = \frac{\epsilon_i^k \mathbf{q}_i(\mathbf{c}^k)^T \mathbf{v}_i^k}{\|\mathbf{v}_i^k\|} = \left[1 + \sum_{j \neq i} \frac{(\xi_{ij}^k)^2 (\lambda_i(\mathbf{c}^k) - \lambda_i^*)^2}{(\xi_{ij}^k)^2 (\lambda_j(\mathbf{c}^k) - \lambda_i^*)^2} \right]^{-\frac{1}{2}}. \tag{3.11}$$

Using an elementary inequality $1 - (1 + t)^{-\frac{1}{2}} \leq t$ for $t \geq 0$ and letting $t = \sum_{j \neq i} \frac{(\xi_{ij}^k)^2 (\lambda_i(\mathbf{c}^k) - \lambda_i^*)^2}{(\xi_{ij}^k)^2 (\lambda_j(\mathbf{c}^k) - \lambda_i^*)^2}$, we obtain

$$1 - \epsilon_i^k \mathbf{q}_i(\mathbf{c}^k)^T \mathbf{p}_i^k = 1 - \left[1 + \sum_{j \neq i} \frac{(\xi_{ij}^k)^2 (\lambda_i(\mathbf{c}^k) - \lambda_i^*)^2}{(\xi_{ij}^k)^2 (\lambda_j(\mathbf{c}^k) - \lambda_i^*)^2} \right]^{-\frac{1}{2}} \leq \sum_{j \neq i} \frac{(\xi_{ij}^k)^2 (\lambda_i(\mathbf{c}^k) - \lambda_i^*)^2}{(\xi_{ij}^k)^2 (\lambda_j(\mathbf{c}^k) - \lambda_i^*)^2}. \tag{3.12}$$

By (3.10) and Lemma 3.1(iii) (with $\mathbf{c} = \mathbf{c}^k$), we have $(\xi_{ii}^k)^2 (\lambda_j(\mathbf{c}^k) - \lambda_i^*)^2 \geq \gamma^2 / 16$. Combining this with (3.9) and (3.12), we arrive at

$$1 - \epsilon_i^k \mathbf{q}_i(\mathbf{c}^k)^T \mathbf{p}_i^k \leq \frac{32(\lambda_i(\mathbf{c}^k) - \lambda_i^*)^2}{\gamma^2}.$$

Since \mathbf{p}_i^k and $\mathbf{q}_i(\mathbf{c}^k)$ are normalized, it follows from Lemma 3.1(i) that

$$\|\epsilon_i^k \mathbf{p}_i^k - \mathbf{q}_i(\mathbf{c}^k)\| = (2(1 - \epsilon_i^k \mathbf{q}_i(\mathbf{c}^k)^T \mathbf{p}_i^k))^{\frac{1}{2}} \leq \frac{8\rho_0}{\gamma} \|\mathbf{c}^k - \mathbf{c}^*\|. \tag{3.13}$$

This and Lemma 3.1(ii) (with $\mathbf{c} = \mathbf{c}^k$) yield

$$\|\epsilon_i^k \mathbf{p}_i^k - \mathbf{q}_i(\mathbf{c}^*)\| \leq \|\epsilon_i^k \mathbf{p}_i^k - \mathbf{q}_i(\mathbf{c}^k)\| + \|\mathbf{q}_i(\mathbf{c}^k) - \mathbf{q}_i(\mathbf{c}^*)\| \leq (1 + 8/\gamma)\rho_0 \|\mathbf{c}^k - \mathbf{c}^*\|. \tag{3.14}$$

Hence assertions (i) and (ii) are proved. \square

Now we present the main result of this paper which shows that the Ulm-like method converges with R-quadratic convergence.

Theorem 3.4. Suppose that $\{\lambda_i^*\}_{i=1}^n$ are distinct and the Jacobian matrix $J(\mathbf{c}^*)$ is invertible. Then there exist $\tau_1, \tau_2 \in (0, +\infty)$, $\delta \in (0, \tau_1)$, and $\mu \in (0, +\infty)$ such that for any $\mathbf{c}^0 \in \mathbf{B}(\mathbf{c}^*, \delta)$ and $B_0 \in \mathbb{R}^{n \times n}$ satisfying (2.11), the sequence $\{\mathbf{c}^k\}$ generated by Algorithm 3 with initial point \mathbf{c}^0 converges to \mathbf{c}^* . Moreover, the following estimates hold for each $k = 0, 1, \dots$

$$\|\mathbf{c}^k - \mathbf{c}^*\| \leq \tau_1 \left(\frac{\delta}{\tau_1}\right)^{2^k} \tag{3.15}$$

and

$$\|I - B_k J_k\| \leq \tau_2 \left(\frac{\delta}{\tau_1}\right)^{2^k}. \tag{3.16}$$

Proof. We write for simplicity,

$$\rho_1 := \frac{\|J(\mathbf{c}^*)^{-1}\|}{1 - H_1 \|J(\mathbf{c}^*)^{-1}\| \delta_1} \quad \text{and} \quad H_2 := n \rho_0^2 (1 + 8/\gamma)^2 \cdot \max_i |\lambda_i^*|.$$

Set

$$\tau_1 := \frac{1}{\rho_1} \min \left\{ \frac{1}{4H_1 + 2\sqrt{2}H_2}, \frac{1}{12H_1} \right\} \quad \text{and} \quad \tau_2 := 4\rho_1 H_1 \tau_1. \tag{3.17}$$

Then $12\rho_1 H_1 \tau_1 \leq 1$ and

$$\tau_2 \leq \frac{1}{3}. \tag{3.18}$$

Take δ and μ such that

$$0 < \delta < \min \left\{ \delta_1, \tau_1, \frac{1}{(2 + 8/\gamma)\rho_0} \right\} \quad \text{and} \quad 0 < \mu \leq 4\rho_1 H_1 \delta. \tag{3.19}$$

We shall show that τ_1, τ_2, δ and μ are as desired.

Let $\mathbf{c}^0 \in \mathbf{B}(\mathbf{c}^*, \delta)$ and $B_0 \in \mathbb{R}^{n \times n}$ satisfy (2.11). It suffices to verify that (3.15), (3.16), and (3.6) hold for each $k = 0, 1, \dots$. We proceed by mathematical induction. Clearly, (3.15) and (3.6) are true for $k = 0$ by assumptions (noting that $\mathbf{p}_i^{-1} = \mathbf{q}_i(\mathbf{c}^0)$). Furthermore, by (2.11), (3.19), and (3.17), we have

$$\|I - B_0 J(\mathbf{c}^0)\| \leq \mu \leq 4\rho_1 H_1 \delta = \tau_2 \frac{\delta}{\tau_1}. \tag{3.20}$$

Thus, thanks to (3.1), the estimate (3.16) holds for $k = 0$. Now assume that (3.15), (3.16) and (3.6) hold for $k = m$, i.e.,

$$\|\mathbf{c}^m - \mathbf{c}^*\| \leq \tau_1 \left(\frac{\delta}{\tau_1}\right)^{2^m}, \quad \|I - B_m J_m\| \leq \tau_2 \left(\frac{\delta}{\tau_1}\right)^{2^m} \tag{3.21}$$

and

$$|\mathbf{q}_i(\mathbf{c}^m)^T \mathbf{p}_i^{m-1}| \geq \frac{1}{2} \quad \text{for each } i \in [1, n]. \tag{3.22}$$

Hence, (3.5) holds for $k = m$ as $\delta < \tau_1$ and $\delta < \delta_1$, i.e.,

$$\|\mathbf{c}^m - \mathbf{c}^*\| \leq \tau_1 \left(\frac{\delta}{\tau_1}\right)^{2^m} \leq \delta < \delta_1. \tag{3.23}$$

Thus, assertions (i)–(iv) in Lemma 3.3 are applicable for $k = m$. In particular,

$$\|\epsilon_i^m \mathbf{p}_i^m - \mathbf{q}_i(\mathbf{c}^*)\| \leq (1 + 8/\gamma)\rho_0 \|\mathbf{c}^m - \mathbf{c}^*\| \quad \text{for each } i \in [1, n], \tag{3.24}$$

$$\|J_m^{-1}\| \leq \frac{\|J(\mathbf{c}^*)^{-1}\|}{1 - H_1 \|J(\mathbf{c}^*)^{-1}\| \delta_1} = \rho_1, \tag{3.25}$$

and

$$\|J_m - J(\mathbf{c}^*)\| \leq H_1 \|\mathbf{c}^m - \mathbf{c}^*\| \leq H_1 \tau_1 \left(\frac{\delta}{\tau_1}\right)^{2^m}. \tag{3.26}$$

Since $\delta < \tau_1$, using (3.21), (3.25), and (3.18), we conclude that

$$\|B_m\| \leq \|B_m J_m\| \cdot \|J_m^{-1}\| \leq (1 + \|I - B_m J_m\|) \|J_m^{-1}\| \leq \rho_1 \left[1 + \tau_2 \left(\frac{\delta}{\tau_1} \right)^{2^m} \right] \leq \sqrt{2} \rho_1. \tag{3.27}$$

Moreover, applying Lemma 3.2 (to $\{\epsilon_i^m \mathbf{p}_i^m\}_{i=1}^n$ in place of $\{\omega_i\}_{i=1}^n$) and (3.24), one has

$$\begin{aligned} \|J_m \mathbf{c}^* - J(\mathbf{c}^*) \mathbf{c}^*\| &\leq 2n \cdot \max_i |\lambda_i^*| \cdot \max_i \|\epsilon_i^m \mathbf{p}_i^m - \mathbf{q}_i(\mathbf{c}^*)\|^2 \\ &\leq 2n \rho_0^2 (1 + 8/\gamma)^2 \cdot \max_i |\lambda_i^*| \cdot \|\mathbf{c}^m - \mathbf{c}^*\|^2 \\ &= 2H_2 \|\mathbf{c}^m - \mathbf{c}^*\|^2, \end{aligned} \tag{3.28}$$

where the equality holds because of the definition of H_2 . Since $\lambda^* = J(\mathbf{c}^*) \mathbf{c}^*$, we note by (2.14) that

$$\begin{aligned} \mathbf{c}^{m+1} - \mathbf{c}^* &= \mathbf{c}^m - \mathbf{c}^* - B_m J_m \mathbf{c}^m + B_m J(\mathbf{c}^*) \mathbf{c}^* \\ &= (I - B_m J_m)(\mathbf{c}^m - \mathbf{c}^*) - B_m (J_m \mathbf{c}^* - J(\mathbf{c}^*) \mathbf{c}^*). \end{aligned} \tag{3.29}$$

It follows from (3.21), (3.27)–(3.29) that

$$\begin{aligned} \|\mathbf{c}^{m+1} - \mathbf{c}^*\| &\leq \|I - B_m J_m\| \cdot \|\mathbf{c}^m - \mathbf{c}^*\| + \|B_m\| \cdot \|J_m \mathbf{c}^* - J(\mathbf{c}^*) \mathbf{c}^*\| \\ &\leq \tau_2 \left(\frac{\delta}{\tau_1} \right)^{2^m} \cdot \tau_1 \left(\frac{\delta}{\tau_1} \right)^{2^m} + \sqrt{2} \rho_1 \cdot 2H_2 \left[\tau_1 \left(\frac{\delta}{\tau_1} \right)^{2^m} \right]^2 \\ &= (\tau_2 + 2\sqrt{2} \rho_1 H_2 \tau_1) \tau_1 \left(\frac{\delta}{\tau_1} \right)^{2^{m+1}} \\ &\leq \tau_1 \left(\frac{\delta}{\tau_1} \right)^{2^{m+1}} \end{aligned} \tag{3.30}$$

(noting that $(\tau_2 + 2\sqrt{2} \rho_1 H_2 \tau_1) \leq 1$ by (3.17)). This proves that (3.15) holds for $k = m + 1$. Consequently,

$$\|\mathbf{c}^{m+1} - \mathbf{c}^*\| \leq \delta < \delta_1, \tag{3.31}$$

that is, (3.5) holds for $k = m + 1$. Furthermore, $\|\mathbf{c}^{m+1} - \mathbf{c}^*\| < \delta_0$ by (3.3). Thus Lemma 3.1 is applicable to getting

$$\|\mathbf{q}_i(\mathbf{c}^{m+1}) - \mathbf{q}_i(\mathbf{c}^*)\| \leq \rho_0 \|\mathbf{c}^{m+1} - \mathbf{c}^*\| \quad \text{for each } i \in [1, n].$$

Fix $i \in [1, n]$. Then (3.23), (3.24), and (3.31) yield

$$\begin{aligned} \|\epsilon_i^m \mathbf{p}_i^m - \mathbf{q}_i(\mathbf{c}^{m+1})\| &\leq \|\epsilon_i^m \mathbf{p}_i^m - \mathbf{q}_i(\mathbf{c}^*)\| + \|\mathbf{q}_i(\mathbf{c}^*) - \mathbf{q}_i(\mathbf{c}^{m+1})\| \\ &\leq (1 + 8/\gamma) \rho_0 \|\mathbf{c}^m - \mathbf{c}^*\| + \rho_0 \|\mathbf{c}^{m+1} - \mathbf{c}^*\| \\ &\leq (2 + 8/\gamma) \rho_0 \delta \\ &\leq 1 \end{aligned}$$

(cf. (3.19) for the last inequality). Since $\|\epsilon_i^m \mathbf{p}_i^m - \mathbf{q}_i(\mathbf{c}^{m+1})\|^2 = 2(1 - \epsilon_i^m \mathbf{q}_i(\mathbf{c}^{m+1})^T \mathbf{p}_i^m)$, it follows from (3.2) that

$$|\mathbf{q}_i(\mathbf{c}^{m+1})^T \mathbf{p}_i^m| \geq \frac{1}{2}. \tag{3.32}$$

Hence (3.6) holds for $k = m + 1$. It remains to show that (3.16) holds for $k = m + 1$. To do this, note that (3.5) holds for $k = m + 1$ as claimed earlier. Applying Lemma 3.3(iii) (with $k = m + 1$) and (3.30) (noting that $\delta < \tau_1$), we have

$$\|J_{m+1} - J(\mathbf{c}^*)\| \leq H_1 \|\mathbf{c}^{m+1} - \mathbf{c}^*\| \leq H_1 \tau_1 \left(\frac{\delta}{\tau_1} \right)^{2^{m+1}} \leq H_1 \tau_1 \left(\frac{\delta}{\tau_1} \right)^{2^m}.$$

This together with (3.26) implies that

$$\|J_{m+1} - J_m\| \leq \|J_{m+1} - J(\mathbf{c}^*)\| + \|J_m - J(\mathbf{c}^*)\| \leq 2H_1 \tau_1 \left(\frac{\delta}{\tau_1} \right)^{2^m}. \tag{3.33}$$

Recalling that $B_{m+1} = 2B_m - B_m J_{m+1} B_m$, we have

$$I - B_{m+1} J_{m+1} = I - 2B_m J_{m+1} + B_m J_{m+1} B_m J_{m+1} = (I - B_m J_{m+1})^2.$$

It follows that

$$\begin{aligned} \|I - B_{m+1} J_{m+1}\| &\leq (\|I - B_m J_m\| + \|B_m\| \cdot \|J_{m+1} - J_m\|)^2 \\ &\leq 2\|I - B_m J_m\|^2 + 2\|B_m\|^2 \cdot \|J_{m+1} - J_m\|^2. \end{aligned} \tag{3.34}$$

Combining this together with (3.21), (3.27), and (3.33), one has

$$\|I - B_{m+1} J_{m+1}\| \leq 2 \left[\tau_2 \left(\frac{\delta}{\tau_1} \right)^{2^m} \right]^2 + 2(\sqrt{2}\rho_1)^2 \left[2H_1 \tau_1 \left(\frac{\delta}{\tau_1} \right)^{2^m} \right]^2 = 3\tau_2^2 \left(\frac{\delta}{\tau_1} \right)^{2^{m+1}},$$

where the equality holds because $\tau_2 = 4\rho_1 H_1 \tau_1$. Then, by (3.18),

$$\|I - B_{m+1} J_{m+1}\| \leq \tau_2 \left(\frac{\delta}{\tau_1} \right)^{2^{m+1}}. \tag{3.35}$$

Thus, (3.16) is proved for $k = m + 1$ and the proof is complete. \square

4. Numerical experiments

In this section, we illustrate the convergence performance of the Ulm-like method on two examples. For comparison, the inexact Newton-like method in [5] is also tested. In particular, an example for which the Ulm-like method converges but not the inexact Newton-like method is provided. In the first example, we consider the inverse Toeplitz eigenvalue problem which was considered in [1,5,27,31].

Example 4.1. We use Toeplitz matrices as our $\{A_i\}_{i=1}^n$ in (1.1):

$$A_1 = I, \quad A_2 = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 1 & 0 & 1 & \ddots & \vdots \\ 0 & 1 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 0 & 1 \\ 0 & \dots & 0 & 1 & 0 \end{pmatrix}, \quad \dots, \quad A_n = \begin{pmatrix} 0 & 0 & \dots & 0 & 1 \\ 0 & \ddots & \ddots & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & \ddots & \ddots & 0 \\ 1 & 0 & \dots & 0 & 0 \end{pmatrix}.$$

Thus $A(\mathbf{c})$ is a symmetric Toeplitz matrix with the first column equal to \mathbf{c} .

All our tests were done in Matlab. In [5], the inexact Newton-like method was tried on 60-by-60 matrices. Here we construct ten 120-by-120 test problems. For each test problem, we first generate \mathbf{c}^* with entries randomly chosen between 0 and 10. Then we compute the eigenvalues $\{\lambda_i^*\}_{i=1}^n$ of $A(\mathbf{c}^*)$ as the prescribed eigenvalues. Since both algorithms are locally convergent, the initial guess \mathbf{c}^0 is formed by chopping the components of \mathbf{c}^* to three decimal places. For both algorithms, the stopping tolerance for the outer (Newton) iterations is 10^{-10} .

Linear systems (2.7) and (2.9) are all solved iteratively by the QMR method (cf. [1,5,12]) using the Matlab-provided QMR function. At the $(k + 1)$ th iteration, we use \mathbf{v}_i^k as the initial guess of the inverse power equations (2.7), and \mathbf{c}^k as the initial guess of the approximate Jacobian equation (2.9). The stopping tolerances for systems (2.7) and (2.9) are given as in the equations. We also set the maximum number of iterations allowed to 400 for all the inner iterations. To speed up the convergence, we use the Matlab-provided Modified ILU (MILU) preconditioner: LUINC(A, [drop-tolerance, 1, 1, 1]) which is one of the most versatile preconditioners for unstructured matrices [1,9,17]. The drop-tolerance we use here is 0.01.

The convergence performances of Algorithms 2 and 3 are illustrated in Table 1, where “ite.” represents the averaged total numbers of outer iterations on ten test problems. Note by (2.11) that B_0 is an approximation to $J(\mathbf{c}^0)^{-1}$. Here, we take $B_0 = J(\mathbf{c}^0)^{-1}$ for the Ulm-like method. Since the inexact Newton-like method converges with a convergence rate β , we present its convergence performances with large β . From this table, we can see that the outer iteration number of the Ulm-like method required for convergence is comparable to that of the inexact Newton-like method with $\beta \geq 1.6$. However, it should be noted that, by computing approximations to the inverse of Jacobian matrices, the Ulm-like method avoids solving the approximate Jacobian equation in each outer iteration. This will be very attractive when the approximate Jacobian matrix is ill-conditioned or the approximate Jacobian equation (2.9) is difficult to solve. Moreover, when the size n is large, we can obtain the sequence $\{B_k\}$ by parallel computation which can further improve the computational efficiency.

To further illustrate the convergence performance of the Ulm-like method, Table 2 gives the averaged values of $\|\mathbf{c}^k - \mathbf{c}^*\|$ and the averaged total numbers of outer iterations of the Ulm-like method with different μ . Here “ite.” is the same as in

Table 1
 $\|\mathbf{c}^k - \mathbf{c}^*\|$ and ite. of Algorithms 2 and 3.

k	Algorithm 2				Algorithm 3
	$\beta = 1.5$	$\beta = 1.6$	$\beta = 1.8$	$\beta = 2$	
0	6.94e-3	6.94e-3	6.94e-3	6.94e-3	6.94e-3
1	8.00e-4	8.00e-4	8.00e-4	8.00e-4	8.00e-4
2	8.58e-7	8.48e-7	8.40e-7	8.40e-7	1.10e-6
3	5.74e-11	1.07e-11	9.12e-12	9.60e-12	2.21e-11
4	4.60e-13	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00
ite.	3.1	3.0	3.0	3.0	3.0

Table 2
 $\|\mathbf{c}^k - \mathbf{c}^*\|$ and ite. of Algorithm 3 with different μ .

k	$\mu = 1e-1$	$\mu = 1e-2$	$\mu = 1e-3$	$\mu = 1e-4$	$\mu = 0$
0	6.94e-3	6.94e-3	6.94e-3	6.94e-3	6.94e-3
1	8.00e-4	8.00e-4	8.00e-4	8.00e-4	8.00e-4
2	4.31e-5	1.46e-6	1.08e-6	1.10e-6	1.10e-6
3	2.14e-8	3.21e-11	2.22e-11	2.53e-11	2.21e-11
4	9.89e-12	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00
ite.	4.0	3.0	3.0	3.0	3.0

Table 1. From Table 2, we can see that, for the Ulm-like method, the convergence performance in the case of $B_0 \approx J(\mathbf{c}^0)^{-1}$ is comparable to that in the case of $B_0 = J(\mathbf{c}^0)^{-1}$.

Below we consider an example of non-Toeplitz matrices, which illustrates that even if the condition numbers are not very large, the inexact Newton-like method fails to converge; while the Ulm-like method still converges.

Example 4.2. This is an inverse problem with $n = 6$. Define

$$A_0 := \mathbf{0}, \quad A_1 := \frac{1}{m_1} \mathbf{e}_1 \mathbf{e}_1^T,$$

$$A_k := \left(\frac{1}{\sqrt{m_1}} \mathbf{e}_1 - \frac{1}{\sqrt{m_k}} \mathbf{e}_k \right) \left(\frac{1}{\sqrt{m_1}} \mathbf{e}_1 - \frac{1}{\sqrt{m_k}} \mathbf{e}_k \right)^T, \quad k = 2, 3, \dots, 6,$$

where $m_1 = 2, m_2 = m_3 = m_4 = m_5 = m_6 = 0.2$. Now

$$\lambda^* = (-310.2490, -249.2218, -28.08413, 113.3087, 218.7351, 487.9554)^T.$$

Then,

$$\mathbf{c}^* = (-83.47955, -53.82911, 89.13261, 40.82639, -47.78696, 21.50871)^T.$$

In this example, we adopt the following four initial points:

- (a) $\mathbf{c}^0 = (-77.95824, -62.08697, 96.54128, 40.10535, -44.33137, 20.79310)^T$;
- (b) $\mathbf{c}^0 = (-76.86213, -63.46336, 95.28928, 41.39452, -42.24157, 17.37889)^T$;
- (c) $\mathbf{c}^0 = (-78.58345, -65.97678, 97.83621, 43.47844, -49.26789, 23.67335)^T$;
- (d) $\mathbf{c}^0 = (-85.47863, -67.28566, 80.28746, 35.38552, -45.45096, 23.47528)^T$.

Here we take $B_0 = J(\mathbf{c}^0)^{-1}$ for the Ulm-like method. For both algorithms, the stopping tolerance for the outer (Newton) iterations is 10^{-10} . Table 3 displays the error of $\|\mathbf{c}^k - \mathbf{c}^*\|$ and the condition numbers $\kappa_2(J_k)$ of J_k for the above four initial points \mathbf{c}^0 , where “ite.” represents the number of outer iterations, and “*” denotes the corresponding algorithm fails to converge, respectively. We see from Table 3 that for these choices of the initial points, the Ulm-like method converges but not the inexact Newton-like method.

Acknowledgements

We would like to thank Professor Z.J. Bai for him to provide us Example 4.2 and to thank two anonymous referees for their valuable comments.

Table 3
 $\|c^k - c^*\|$, ite. and $\kappa_2(J_k)$ of Algorithms 2 and 3.

ini.	k	Algorithm 2				$\kappa_2(J_k)$	Algorithm 3	
		$\ c^k - c^*\ $					$\ c^k - c^*\ $	$\kappa_2(J_k)$
		$\beta = 1.5$	$\beta = 1.6$	$\beta = 1.8$	$\beta = 2.0$			
(a)	0	1.29e+1	1.29e+1	1.29e+1	1.29e+1	1.64e+1	1.29e+1	1.64e+1
	1	1.10e+0	1.10e+0	1.10e+0	1.10e+0	1.64e+1	1.10e+0	1.64e+1
	2	1.10e+0	1.10e+0	1.10e+0	1.10e+0	1.64e+1	1.49e-1	1.64e+1
	3	1.10e+0	1.10e+0	1.10e+0	1.10e+0	1.64e+1	6.70e-3	1.64e+1
	4	1.10e+0	1.10e+0	1.10e+0	1.10e+0	1.64e+1	2.43e-5	1.64e+1
	5	1.10e+0	1.10e+0	1.10e+0	1.10e+0	1.64e+1	4.70e-10	1.64e+1
	6	1.10e+0	1.10e+0	1.10e+0	1.10e+0	1.64e+1	1.20e-13	
	7	1.10e+0	1.10e+0	1.10e+0	1.10e+0	1.64e+1	0.00	
	8	1.10e+0	1.10e+0	1.10e+0	1.10e+0	1.64e+1	0.00	
	9	1.10e+0	1.10e+0	1.10e+0	1.10e+0	1.64e+1	0.00	
	10	1.10e+0	1.10e+0	1.10e+0	1.10e+0	1.64e+1	0.00	
ite.	*	*	*	*		6		
(b)	0	1.49e+1	1.49e+1	1.49e+1	1.49e+1	1.63e+1	1.49e+1	1.63e+1
	1	1.39e+0	1.39e+0	1.39e+0	1.39e+0	1.64e+1	1.39e+0	1.64e+1
	2	1.39e+0	1.39e+0	1.39e+0	1.39e+0	1.64e+1	2.02e-1	1.64e+1
	3	1.39e+0	1.39e+0	1.39e+0	1.39e+0	1.64e+1	1.10e-2	1.64e+1
	4	1.39e+0	1.39e+0	1.39e+0	1.39e+0	1.64e+1	6.05e-5	1.64e+1
	5	1.39e+0	1.39e+0	1.39e+0	1.39e+0	1.64e+1	2.79e-9	1.64e+1
	6	1.39e+0	1.39e+0	1.39e+0	1.39e+0	1.64e+1	8.00e-14	1.64e+1
	7	1.39e+0	1.39e+0	1.39e+0	1.39e+0	1.64e+1	0.00	
	8	1.39e+0	1.39e+0	1.39e+0	1.39e+0	1.64e+1	0.00	
	9	1.39e+0	1.39e+0	1.39e+0	1.39e+0	1.64e+1	0.00	
	10	1.39e+0	1.39e+0	1.39e+0	1.39e+0	1.64e+1	0.00	
ite.	*	*	*	*		6		
(c)	0	1.62e+1	1.62e+1	1.62e+1	1.62e+1	1.64e+1	1.62e+1	1.64e+1
	1	8.97e-1	8.97e-1	8.97e-1	8.97e-1	1.64e+1	8.97e-1	1.64e+1
	2	8.97e-1	8.97e-1	8.97e-1	8.97e-1	1.64e+1	1.11e-1	1.64e+1
	3	8.97e-1	8.97e-1	8.97e-1	8.97e-1	1.64e+1	3.98e-3	1.64e+1
	4	8.97e-1	8.97e-1	8.97e-1	8.97e-1	1.64e+1	8.96e-6	1.64e+1
	5	8.97e-1	8.97e-1	8.97e-1	8.97e-1	1.64e+1	6.63e-11	
	6	8.97e-1	8.97e-1	8.97e-1	8.97e-1	1.64e+1	0.00	
	8	8.97e-1	8.97e-1	8.97e-1	8.97e-1	1.64e+1	0.00	
	9	8.97e-1	8.97e-1	8.97e-1	8.97e-1	1.64e+1	0.00	
	10	8.97e-1	8.97e-1	8.97e-1	8.97e-1	1.64e+1	0.00	
	ite.	*	*	*	*		5	
(d)	0	1.74e+1	1.74e+1	1.74e+1	1.74e+1	1.63e+1	1.74e+1	1.63e+1
	1	1.33e+0	1.33e+0	1.33e+0	1.33e+0	1.64e+1	1.33e+0	1.64e+1
	2	1.33e+0	1.33e+0	1.33e+0	1.33e+0	1.64e+1	1.90e-1	1.64e+1
	3	1.33e+0	1.33e+0	1.33e+0	1.33e+0	1.64e+1	9.93e-3	1.64e+1
	4	1.33e+0	1.33e+0	1.33e+0	1.33e+0	1.64e+1	5.02e-5	1.64e+1
	5	1.33e+0	1.33e+0	1.33e+0	1.33e+0	1.64e+1	1.94e-9	1.64e+1
	6	1.33e+0	1.33e+0	1.33e+0	1.33e+0	1.64e+1	8.00e-14	
	7	1.33e+0	1.33e+0	1.33e+0	1.33e+0	1.64e+1	0.00	
	8	1.33e+0	1.33e+0	1.33e+0	1.33e+0	1.64e+1	0.00	
	9	1.33e+0	1.33e+0	1.33e+0	1.33e+0	1.64e+1	0.00	
	10	1.33e+0	1.33e+0	1.33e+0	1.33e+0	1.64e+1	0.00	
ite.	*	*	*	*		6		

References

- [1] Z.J. Bai, R.H. Chan, B. Morini, An inexact Cayley transform method for inverse eigenvalue problems, *Inverse Problems* 20 (2004) 1675–1689.
- [2] Z.J. Bai, X.Q. Jin, A Note on the Ulm-like Method for Inverse Eigenvalue Problems, *Recent Advances in Scientific Computing and Matrix Analysis*, International Press, 2011.
- [3] P.J. Brussard, P.W. Glaudemans, *Shell Model Applications in Nuclear Spectroscopy*, Elsevier, New York, 1977.
- [4] C.I. Byrnes, Pole placement by output feedback, three decades of mathematics systems theory, *Lecture Notes in Control and Inform. Sci.* 135 (1989) 31–78.
- [5] R.H. Chan, H.L. Chung, S.F. Xu, The inexact Newton-like method for inverse eigenvalue problem, *BIT* 43 (2003) 7–20.
- [6] R.H. Chan, S.F. Xu, H.M. Zhou, On the convergence of a quasi-Newton method for inverse eigenvalue problems, *SIAM J. Numer. Anal.* 36 (1999) 436–441.
- [7] M.T. Chu, Inverse eigenvalue problems, *SIAM Rev.* 40 (1998) 1–39.
- [8] M.T. Chu, G.H. Golub, Structured inverse eigenvalue problems, *Acta Numer.* 11 (2002) 1–71.
- [9] T. Dupont, R.P. Kendall, H.H. Rachford Jr., An approximate factorization procedure for solving self-adjoint elliptic difference equations, *SIAM J. Numer. Anal.* 5 (1968) 559–573.
- [10] S. Elhay, Y.M. Ram, An affine inverse eigenvalue problem, *Inverse Problems* 18 (2002) 455–466.

- [11] J.A. Ezquerro, M.A. Hernández, The Ulm method under mild differentiability conditions, *Numer. Math.* 109 (2008) 193–207.
- [12] R.W. Freund, N.M. Nachtigal, QMR: a quasi-minimal residual method for non-Hermitian linear systems, *Numer. Math.* 60 (1991) 315–339.
- [13] S. Friedland, J. Nocedal, M.L. Overton, The formulation and analysis of numerical methods for inverse eigenvalue problems, *SIAM J. Numer. Anal.* 24 (1987) 634–667.
- [14] A. Galperin, Z. Waksman, Ulm's method under regular smoothness, *Numer. Funct. Anal. Optim.* 19 (1998) 285–307.
- [15] G.M.L. Gladwell, Inverse problems in vibration, *Appl. Mech. Rev.* 39 (1986) 1013–1018.
- [16] G.M.L. Gladwell, Inverse problems in vibration II, *Appl. Mech. Rev.* 49 (1996) 25–34.
- [17] I. Gustafsson, A class of first order factorizations, *BIT* 18 (1978) 142–156.
- [18] J.M. Gutiérrez, M.A. Hernández, N. Romero, A note on a modification of Moser's method, *J. Complexity* 24 (2008) 185–197.
- [19] O. Hald, On discrete and numerical Sturm–Liouville problems, PhD dissertation, Dept. Mathematics, New York University, New York, 1970.
- [20] O. Hald, On a Newton–Moser type method, *Numer. Math.* 23 (1975) 411–426.
- [21] K.T. Joseph, Inverse eigenvalue problem in structural design, *AIAA J.* 30 (1992) 2890–2896.
- [22] W.N. Kublanovskaja, On an approach to the solution of the inverse eigenvalue problem, *Zap. Nauch. Sem. Leningrad. Otdel. Mat. Inst. Im. V.A. Steklova Akad. Nauk SSSR* (1970) 138–149.
- [23] N. Li, A matrix inverse eigenvalue problem and its application, *Linear Algebra Appl.* 266 (1997) 143–152.
- [24] J. Moser, *Stable and Random Motions in Dynamical Systems*, Princeton University Press, Princeton, NJ, 1973.
- [25] M. Müller, An inverse eigenvalue problem: computing B-stable Runge–Kutta methods having real poles, *BIT* 32 (1992) 676–688.
- [26] R.L. Parker, K.A. Whaler, Numerical methods for establishing solutions to the inverse problem of electromagnetic induction, *J. Geophys. Res.* 86 (1981) 9574–9584.
- [27] J. Peinado, A.M. Vidal, *A New Parallel Approach to the Toeplitz Inverse Eigenproblem Using Newton-Like Methods*, Lecture Notes in Comput. Sci., Springer, Berlin, 1981.
- [28] H. Petzeltova, Remark on Newton–Moser type method, *Commentat. Math. Univ. Carol.* 21 (1980) 719–725.
- [29] F.A. Potra, V. Pták, Nondiscrete induction and an inversion-free modification of Newton's method, *Čas. Pro Pěstování Mat.* 108 (1983) 333–341.
- [30] M.S. Ravi, J. Rosenthal, X.A. Wang, On decentralized dynamic pole placement and feedback stabilization, *IEEE Trans. Automat. Control.* 40 (1995) 1603–1614.
- [31] W.F. Trench, Numerical solution of the inverse eigenvalue problem for real symmetric Toeplitz matrices, *SIAM J. Sci. Comput.* 18 (1997) 1722–1736.
- [32] S. Ulm, On iterative methods with successive approximation of the inverse operator, *Izv. Akad. Nauk Est. SSR* 16 (1967) 403–411.
- [33] S.F. Xu, *An Introduction to Inverse Algebraic Eigenvalue Problems*, Peking University Press, Beijing, 1998.
- [34] E.J. Zehnder, A remark about Newton's method, *Commun. Pure Appl. Math.* 27 (1974) 361–366.