A Note on the Ulm-like Method for Inverse Eigenvalue Problems

Zheng-jian Bai* Xiao-qing Jin†

Abstract

A Ulm-like method is proposed in [13] for solving inverse eigenvalue problems with distinct given eigenvalues. The Ulm-like method avoids solving the Jacobian equations used in Newton-like methods and is shown to be quadratically convergent in the root sense. However, the numerical experiments in [3] only show that the Ulm-like method is comparable to the inexact Newton-like method. In this short note, we give a numerical example to show that the Ulm-like method is better than the inexact Newton-like method in terms of convergence neighborhoods.

Keywords. Inverse eigenvalue problem, Ulm-like method, inexact Newton-like method.

AMS subject classifications. 65F18, 65F10, 65F15.

1 Introduction

Let $A_1, A_2, \ldots, A_n$ be n real symmetric n-by-n matrices. For any $\mathbf{c} = (c_1, \ldots, c_n)^T \in \mathbb{R}^n$, let $\{\lambda_i(\mathbf{c})\}_{i=1}^n$ denote the eigenvalues of the matrix

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

where $\lambda_1(\mathbf{c}) \leq \lambda_2(\mathbf{c}) \leq \cdots \leq \lambda_n(\mathbf{c})$. In this note, we consider the inverse eigenvalue problem (IEP) defined as follows:

IEP: Given n real numbers $\lambda_1^* \leq \cdots \leq \lambda_n^*$, find a vector $\mathbf{c}^* \in \mathbb{R}^n$ such that $\lambda_i(\mathbf{c}^*) = \lambda_i^*$ for $i = 1, \ldots, n$.

The inverse eigenvalue problem arises in many applications, such as the solution of inverse Sturm-Liouville problems, nuclear and molecular spectroscopy, applied mechanics and structure design, etc. For applications, theoretical and algorithmic aspects of general inverse eigenvalue problems, see [4, 5, 6, 8, 9, 10, 11, 14].

Assume that the given eigenvalues are distinct and there exists a solution $\mathbf{c}^*$ for the IEP. Then there exists a neighborhood $\mathcal{N}(\mathbf{c}^*)$ of $\mathbf{c}^*$ where $\lambda_i(\mathbf{c})$ are distinct and $\mathbf{f}(\mathbf{c})$ is

*School of Mathematical Sciences, Xiamen University, Xiamen 361005, P. R. China (zjbai@xmu.edu.cn).
The research of this author is partially supported by NCETXMU and SRF for ROCS and SEM.
†Department of Mathematics, University of Macau, Macao (xqjin@umac.mo).
The research of this author is supported by the grant UL020/08-Y2/MAT/JXQ01/FST from University of Macau.

differentiable [14]. In this neighborhood $\mathcal{N}(\mathbf{c}^*)$, the IEP can be written as a nonlinear equation

$$\mathbf{f}(\mathbf{c}) \equiv (\lambda_1(\mathbf{c}) - \lambda_1^*, \dots, \lambda_n(\mathbf{c}) - \lambda_n^*)^T = \mathbf{0}, \quad \forall\, \mathbf{c} \in \mathcal{N}(\mathbf{c}^*). \tag{1}$$

It is well-known [8, 14] that Newton-like methods can be employed for solving the IEP based on (1). For the inexact Newton-like approaches, see for instance [1, 3]. In each outer Newton iteration, however, we need to solve the (approximate) Jacobian equation, which may cause the unstability problem if the Jacobian matrix is ill-conditioned.

In [13], a Ulm-like method is proposed for the IEP. The advantage of the Ulm-like method is replacing the solution of the (approximate) Jacobian equations by computing the product of matrices. Therefore, the Ulm-like method avoids the unstability problem caused by the possible ill-conditioning in solving the (approximate) Jacobian equations. The Ulm-like method has a root-quadratically convergence rate under the assumption of the distinction of given eigenvalues.

However, the numerical tests in [13] only show that the Ulm-like method is comparable to the inexact Newton-like method in terms of outer Newton iterations. In this note, we present an example to show that the Ulm-like method is better than the inexact Newton-like method, when the initial guess $\mathbf{c}^0$ is far from the solution $\mathbf{c}^*$ to the IEP.

2 Inexact Newton-Like Method and Ulm-like Method

We first recall the inexact Newton-like method and the Ulm-like method respectively for solving the IEP. Let $\|\cdot\|$ denote the Euclidean norm in $\mathbb{R}^n$ and the corresponding induced matrix norm in $\mathbb{R}^{n\times n}$, i.e.,

$$\|A\| \equiv \max\left\{\frac{\|A\mathbf{x}\|}{\|\mathbf{x}\|} : \mathbf{x} \in \mathbb{R}^n \text{ with } \mathbf{x} \neq \mathbf{0}\right\}.$$

Suppose $\mathbf{c}^*$ is the solution of the IEP. Let $\lambda_1(\mathbf{c}) \le \lambda_2(\mathbf{c}) \le \cdots \le \lambda_n(\mathbf{c})$ be the eigenvalues of the matrix $A(\mathbf{c})$ and $\mathbf{p}_k(\mathbf{c})$ be the normalized eigenvector corresponding to $\lambda_k(\mathbf{c})$. The matrix $J(\mathbf{c})$ is defined by

$$[J(\mathbf{c})]_{ij} := \mathbf{p}_i(\mathbf{c})^T A_j \mathbf{p}_i(\mathbf{c}),$$

for $i, j = 1, 2, \dots, n$. Let

$$\boldsymbol{\lambda}^* = (\lambda_1^*, \lambda_2^*, \dots, \lambda_n^*)^T,$$

where $\lambda_1^* \le \lambda_2^* \le \cdots \le \lambda_n^*$.

The inexact Newton-like algorithm [3] is given as follows.

Algorithm I: Inexact Newton-like Method

1. *Given* $\mathbf{c}^0$, *compute the orthonormal eigenvectors* $\{\mathbf{p}_i^0 = \mathbf{p}_i(\mathbf{c}^0)\}_{i=1}^n$ *of* $A(\mathbf{c}^0)$ *and solve* $\mathbf{c}^1$ *by*

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

2. *For* $k = 0, 1, 2, \dots,$ *until convergence, do:*

   (a) *Solve* $\mathbf{v}_i^k$ *inexactly by the one-step inverse power method:*

$$\big(A(\mathbf{c}^k) - \lambda_i^* I\big)\mathbf{v}_i^k = \mathbf{p}_i^{k-1} + \mathbf{t}_i^k, \quad i = 1, 2, \dots, n, \tag{2}$$

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

$$\|\mathbf{t}_i^k\| \le \frac{1}{4}, \quad i = 1, 2, \ldots, n.$$

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

$$\mathbf{p}_i^k = \frac{\mathbf{v}_i^k}{\|\mathbf{v}_i^k\|}, \quad i = 1, 2, \ldots, n.$$

(c) *Form the approximate Jacobian matrix* J_k*:*

$$[J_k]_{ij} = (\mathbf{p}_i^k)^T A_j \mathbf{p}_i^k, \quad i, j = 1, 2, \ldots, n. \tag{3}$$

(d) *Solve* $\mathbf{c}^{k+1}$ *inexactly from the approximate Jacobian equation:*

$$J_k \mathbf{c}^{k+1} = \boldsymbol{\lambda}^* + \mathbf{r}^k \tag{4}$$

until the residual $\mathbf{r}^k$ *satisfies*

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

Under the assumptions that the given eigenvalues are distinct and the Jacobian matrix $J(\mathbf{c}^*)$ is nonsingular, it was proved in [3] that the method converges locally with a convergence rate β.

Next, we present the Ulm-like method for the IEP.

Algorithm II: Ulm-like Method

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

$$\|I - B_0 J(\mathbf{c}^0)\| \le \delta,$$

where δ *is a positive constant. Compute the orthonormal eigenvectors* $\{\mathbf{p}_i^0 = \mathbf{p}_i(\mathbf{c}^0)\}_{i=1}^n$ *of* $A(\mathbf{c}^0)$ *and solve* $\mathbf{c}^1$ *by*

$$\mathbf{c}^1 = \mathbf{c}^0 - B_0\big(J(\mathbf{c}^0)\mathbf{c}^0 - \boldsymbol{\lambda}^*\big).$$

2. *For* $k = 0, 1, 2, \ldots,$ *until convergence, do:*

(a) *Solve* $\mathbf{v}_i^k$ *inexactly by the one-step inverse power method:*

$$\big(A(\mathbf{c}^k) - \lambda_i^* I\big)\mathbf{v}_i^k = \mathbf{p}_i^{k-1} + \mathbf{t}_i^k, \quad i = 1, 2, \ldots, n, \tag{5}$$

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

$$\|\mathbf{t}_i^k\| \le \frac{1}{4}, \quad i = 1, 2, \ldots, n.$$

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

$$\mathbf{p}_i^k = \frac{\mathbf{v}_i^k}{\|\mathbf{v}_i^k\|}, \quad i = 1, 2, \ldots, n.$$

(c) *Form the approximate Jacobian matrix* J_k*:*

$$[J_k]_{ij} = (\mathbf{p}_i^k)^T A_j \mathbf{p}_i^k, \quad i, j = 1, 2, \ldots, n.$$

(d) *Compute 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

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

Under the assumptions that the given eigenvalues are distinct and the Jacobian matrix $J(\mathbf{c}^*)$ is nonsingular, the Ulm-like method converges quadratically in the root sense [13].

3 Numerical Tests

In this section, we compare the numerical performance of Algorithm I with that of Algorithm II on an example below. Our goal is to illustrate the advantage of the Ulm-like method over the inexact Netwon-like method, when the approximate Jacobian equation (4) is ill-conditioned.

Since the given eigenvalues are assumed to be distinct, the function $f(\mathbf{c})$ in (1) is differentiable, $J(\mathbf{c})$ is Lipsschitz continuous in the neighborhood $\mathcal{N}(\mathbf{c}^*)$ of $\mathbf{c}^*$, and $J(\mathbf{c}^*)$ is nonsingular [14]. For $\mathbf{c} \in \mathcal{N}(\mathbf{c}^*)$, by (see [12])

$$\frac{\partial^2 \lambda_i}{\partial c_k \, \partial c_j} = 2 \sum_{\substack{m=1 \\ m \neq i}} \frac{[\mathbf{q}_m \mathbf{c}^T A_k \mathbf{q}_i \mathbf{c}][\mathbf{q}_m \mathbf{c}^T A_j \mathbf{q}_i \mathbf{c}]}{\lambda_i(\mathbf{c}) - \lambda_m(\mathbf{c})},$$

it follows that the Lipschitz constant may be very large if some of the eigenvalues λ_i^* are close to each other. Then, for the inexact Newton-like method, the approximate Jacobian J_k defined in (3) may be ill-conditioned and the convergence neighborhood $\mathcal{N}(\mathbf{c}^*)$ may be very small [8].

Example 3.1 *Given* $B = I_8 + VV^T$*, where*

$$V = \begin{bmatrix} 1 & -1 & -3 & -5 & -6 \\ 1 & 1 & -2 & -5 & -17 \\ 1 & -1 & -1 & 5 & 18 \\ 1 & 1 & 1 & 2 & 0 \\ 1 & -1 & 2 & 0 & 1 \\ 1 & 1 & 3 & 0 & -1 \\ 2.5 & .2 & .3 & .5 & .6 \\ 2 & -.2 & .3 & .5 & .8 \end{bmatrix}_{8\times 5}$$

Define A_k *by*

$$A_k = b_{kk}\mathbf{e}_k\mathbf{e}_k^T + \sum_{j=1}^{k-1} b_{kj}(\mathbf{e}_k\mathbf{e}_j^T + \mathbf{e}_j\mathbf{e}_k^T), \quad k = 1, \ldots, 8.$$

Note that if

$$\hat{\mathbf{c}} = (1, 1, 1, 1, 1, 1, 1, 1)^T,$$

then

$$A(\hat{\mathbf{c}}) = B = I_8 + VV^T,$$

whose eigenvalues are given by

$$\hat{\boldsymbol{\lambda}} = (1, 1, 1, 2.1208, 9.2189, 17.2814, 35.7082, 722.6808)^T.$$

Now, suppose that

$$\begin{aligned}\mathbf{c}^* &= (1.000438903816714, 1.000656447518457, 1.000913442705718, 1.000231554995865,\\ &\quad 0.999744815493349, 0.999113996722789, 1.000942919907134, 0.999654879193127)^T.\end{aligned}$$

Then, we have

$$\begin{aligned}\boldsymbol{\lambda}^* &= (0.9793644297787, 0.9976265969314, 1.0039322015831, 2.1258971800068,\\ &\quad 9.2125235810642, 17.2782020459764, 35.6897669639946, 723.2816411319387)^T.\end{aligned}$$

In Example 3.1, we observe that the first three eigenvalues λ_1, λ_2, λ_3 are distinct but very close to each other:

$$|\lambda_2 - \lambda_1| = 1.8262e-002, \qquad |\lambda_3 - \lambda_2| = 6.3056e-003.$$

Thus, the approximate Jacobian J_k in (3) may be ill-conditioned and the convergence neighborhood $\mathcal{N}(\mathbf{c}^*)$ becomes small.

To illustrate the numerical performance of the Ulm-like method, we compare the Ulm-like method with the inexact Newton-like method for Example 3.1. The initial point $\mathbf{c}^0$ is chosen by

$$\mathbf{c}^0 = \frac{\texttt{floor}(\phi \times 10^\psi \cdot \mathbf{c}^*)}{\phi \times 10^\psi},$$

where the `Matlab`-routine function `floor`($\mathbf{c}$) rounds the elements of a vector $\mathbf{c}$ to the nearest integers towards minus infinity. We consider the following four cases: (a) $\phi = 3$ and $\psi = 1$; (b) $\phi = 4$ and $\psi = 1$; (c) $\phi = 1$ and $\psi = 2$; (d) $\phi = 1$ and $\psi = 4$.

The linear systems (2), (4), and (5) are solved iteratively by the QMR method [7] as in [13, 1, 3] 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) and (5), and $\mathbf{c}^k$ as the initial guess of the approximate Jacobian equation (4). We also set the maximum number of iterations allowed to 400 for all 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 used for unconstructed matrices [13, 1, 2]. The drop-tolerance we use here is 0.01. The stopping tolerance for the outer Newton iterations of Algorithms I and II are set to be 10^{-10} (we set the maximum number of outer iterations to be 10).

Table 1 displays the error of $\|\mathbf{c}^k - \mathbf{c}^*\|$ and the condition number of $\kappa_2(J_k)$ for different initial guess $\mathbf{c}^0$, where $\mathbf{c}^k$ is the iterant at the kth outer iteration, `it.` denotes the number of required outer iterations, and $\star$ denotes the corresponding algorithm fails to converge. We observe from Table 1 the following facts: When the initial guess $\mathbf{c}^0$ is somewhat far from the solution $\mathbf{c}^*$, the approximate Jacobian equation (4) is increasingly ill-conditioned and the inexact Newton-like method fails to converge, while the Ulm-like method still converges. On the other hand, when the initial guess $\mathbf{c}^0$ is close to $\mathbf{c}^*$ sufficiently, the approximate Jacobian equation (4) is well-conditioned. In this case, the Ulm-like method is comparable to the inexact Newton-like method. This numerical example shows that the Ulm-like method has a larger convergence neighborhood than that of the inexact Newton-like method.

References

[1] Z.-J. Bai, R. H. Chan, and B. Morini, *An inexact Cayley transform method for inverse eigenvalue problems*, Inverse Problems, 20 (2004), pp. 1675–1689.

[2] Z.-J. Bai and S.-F. Xu, *An inexact Newton-type method for inverse singular value problems*, Linear Algebra Appl., 429 (2008), pp. 527–547.

[3] R. H. Chan, H.-L. Chung, and S.-F. Xu, *The inexact Newton-like method for inverse eigenvalue problem*, BIT, 43 (2003), pp. 7–20.

[4] M. T. Chu, *Inverse eigenvalue problems*, SIAM Rev., 40 (1998), pp. 1–39.

[5] M. T. Chu and G. H. Golub, *Structured inverse eigenvalue problems*, Acta Numer., 11 (2002), pp. 1–71.

[6] M. T. Chu and G. H. Golub, *Inverse Eigenvalue Problems: Theory, Algorithms, and Applications*, Oxford University Press, Oxford, 2005.

[7] R. W. Freund and N. M. Nachtigal, *QMR: A quasi-minimal residual method for non-hermitian linear systems*, Numer. Math., 60 (1991), pp. 315–339.

[8] S. Friedland, J. Nocedal, and M. L. Overton, *The formulation and analysis of numerical methods for inverse eigenvalue problems*, SIAM J. Numer. Anal., 24 (1987), pp. 634–667.

[9] G. M. L. Gladwell, *Inverse problems in vibration*, Appl. Mech. Rev., 39 (1986), pp. 1013–1018.

[10] G. M. L. Gladwell, *Inverse problems in vibration, II*, Appl. Mech. Rev., 49 (1996), pp. 25–34.

[11] G. M. L. Gladwell, *Inverse Problems in Vibration*, Kluwer Academic Publishers, Dordrecht, 2004.

[12] P. Lancaster, *On eigenvalues of matrices dependent on a parameter*, Numer. Math., 6 (1964), pp. 377–387.

[13] C. Li, W.-P. Shen, and X.-Q. Jin, *A Ulm-like method for inverse eigenvalue problems*, Numer. Math., to appear.

[14] S.-F. Xu, *An Introduction to Inverse Algebraic Eigenvalue Problems*, Peking University Press and Vieweg Publishing, Beijing, 1998.

Table 1: Numerical results for Example 3.1

(a)

k	Inexact Newton-like Method				Ulm-like Method	
	$\|\mathbf{c}^k - \mathbf{c}^*\|$					
	$\beta = 1.6$	$\beta = 1.8$	$\beta = 2.0$	$\kappa_2(J_k)$	$\|\mathbf{c}^k - \mathbf{c}^*\|$	$\kappa_2(J_k)$
0	$5.6900e-002$	$5.6900e-002$	$5.6900e-002$	$6.2633e+002$	$5.6900e-002$	$6.2633e+002$
1	$1.5972e-002$	$1.5972e-002$	$1.5972e-002$	$2.0217e+003$	$1.5972e-002$	$2.0217e+003$
2	$7.5108e-003$	$7.5108e-003$	$7.5108e-003$	$1.2617e+003$	$2.4305e-003$	$1.1030e+003$
3	$5.2259e-003$	$5.2259e-003$	$5.2259e-003$	$4.8507e+004$	$8.5988e-005$	$1.0936e+003$
4	$8.5108e-002$	$8.5108e-002$	$8.5108e-002$	$1.4192e+003$	$3.7376e-006$	$1.0850e+003$
5	$7.9658e-002$	$7.9658e-002$	$7.9658e-002$	$3.1942e+005$	$3.2939e-009$	$1.0852e+003$
6	$2.7836e+000$	$2.7836e+000$	$2.7836e+000$	$7.4580e+004$	$2.9768e-014$	
7	$2.7836e+000$	$2.7836e+000$	$2.7836e+000$	$5.5743e+006$		
8	$2.7836e+000$	$2.7836e+000$	$2.7836e+000$	$2.1713e+008$		
9	$2.7836e+000$	$2.7836e+000$	$2.7836e+000$	$1.2112e+009$		
10	$2.7836e+000$	$2.7836e+000$	$2.7836e+000$			
it.	$\star$	$\star$	$\star$		6	

(b)

k	Inexact Newton-like Method				Ulm-like Method	
	$\|\mathbf{c}^k - \mathbf{c}^*\|$					
	$\beta = 1.6$	$\beta = 1.8$	$\beta = 2.0$	$\kappa_2(J_k)$	$\|\mathbf{c}^k - \mathbf{c}^*\|$	$\kappa_2(J_k)$
0	$4.2474e-002$	$4.2474e-002$	$4.2474e-002$	$6.3724e+002$	$4.2474e-002$	$6.3724e+002$
1	$1.5893e-002$	$1.5893e-002$	$1.5893e-002$	$1.8435e+003$	$1.5893e-002$	$1.8435e+003$
2	$5.8073e-003$	$5.8073e-003$	$5.8613e-003$	$1.1812e+003$	$2.3494e-003$	$1.1145e+003$
3	$2.7831e-003$	$2.7184e-003$	$2.1950e-003$	$1.4335e+003$	$2.7159e-004$	$1.1052e+003$
4	$2.1579e-004$	$1.8335e-004$	$8.7238e-005$	$1.0798e+003$	$5.7055e-006$	$1.0849e+003$
5	$2.5776e-006$	$1.6465e-006$	$4.2657e-007$	$1.0851e+003$	$7.6267e-009$	$1.0852e+003$
6	$4.6985e-010$	$1.3123e-010$	$8.7389e-012$		$1.6943e-014$	
7	$2.5663e-014$	$1.9658e-014$				
it.	7	7	6		6	

(c)

k	Inexact Newton-like Method				Ulm-like Method	
	$\|\mathbf{c}^k - \mathbf{c}^*\|$					
	$\beta = 1.6$	$\beta = 1.8$	$\beta = 2.0$	$\kappa_2(J_k)$	$\|\mathbf{c}^k - \mathbf{c}^*\|$	$\kappa_2(J_k)$
0	$1.6542e-002$	$1.6542e-002$	$1.6542e-002$	$6.6323e+002$	$1.6542e-002$	$6.6323e+002$
1	$1.6320e-002$	$1.6320e-002$	$1.6320e-002$	$1.7985e+003$	$1.6320e-002$	$1.7985e+003$
2	$5.2114e-003$	$5.2114e-003$	$5.3287e-003$	$1.1766e+003$	$1.3189e-003$	$1.0968e+003$
3	$1.7118e-003$	$1.6561e-003$	$1.4067e-003$	$1.2546e+003$	$2.0367e-004$	$1.0961e+003$
4	$6.2036e-005$	$5.2056e-005$	$3.7954e-005$	$1.0829e+003$	$1.2452e-005$	$1.0857e+003$
5	$2.4555e-007$	$1.4633e-007$	$7.7295e-008$	$1.0852e+003$	$2.9611e-008$	$1.0852e+003$
6	$2.0155e-011$	$9.9290e-013$	$2.4267e-013$		$1.0741e-013$	
it.	7	7	6		6	

(d)

k	Inexact Newton-like Method				Ulm-like Method	
	$\|\mathbf{c}^k - \mathbf{c}^*\|$					
	$\beta = 1.6$	$\beta = 1.8$	$\beta = 2.0$	$\kappa_2(J_k)$	$\|\mathbf{c}^k - \mathbf{c}^*\|$	$\kappa_2(J_k)$
0	$1.1373e-004$	$1.1373e-004$	$1.1373e-004$	$1.0856e+003$	$1.1373e-004$	$1.0856e+003$
1	$2.0352e-007$	$2.0352e-007$	$2.0352e-007$	$1.0852e+003$	$2.0352e-007$	$1.0852e+003$
2	$2.9669e-014$	$2.9669e-014$	$2.9669e-014$		$1.5068e-013$	
it.	2	2	2		2	