Inexact Numerical Methods for Inverse Eigenvalue Problems

Zheng-jian Bai*

November 30, 2006

Abstract

In this paper, we survey some of the latest development in using inexact Newton-like methods for solving inverse eigenvalue problems. These methods require the solutions of nonsymmetric and large linear systems. One can solve the approximate Jacobian equation by iterative methods. However, iterative methods usually oversolve the problem in the sense that they require far more (inner) iterations than is required for the convergence of the Newton-like (outer) iterations. The inexact methods can reduce or minimize the oversolving problem and improve the efficiency. The convergence rates of the inexact methods are superlinear and a good tradeoff between the required inner and outer iterations can be obtained.

AMS Subject Classifications. 65F18, 65F10, 65F15.

1 Introduction

Let $\{A_k\}_{k=1}^n$ be n real symmetric $n \times n$ matrices. For any $\mathbf{c} = (c_1, \ldots, c_n)^T \in \mathbb{R}^n$, we define

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

and denote the eigenvalues of $A(\mathbf{c})$ by $\{\lambda_i(\mathbf{c})\}_{i=1}^n$, where $\lambda_1(\mathbf{c}) \le \lambda_2(\mathbf{c}) \le \cdots \le \lambda_n(\mathbf{c})$. The Inverse Eigenvalue Problem (IEP) is defined as follows:

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

There is a large literature on conditions for the solvability, perturbation analysis and computational methods to the IEP, see for instance [8, 12, 13] and the references therein. Recently Friedland, Nocedal, and Overton [8] have surveyed four fast locally convergent numerical methods for solving the IEP.

For a general nonlinear system $\mathbf{g}(\mathbf{x}) = \mathbf{0}$, the classical locally convergent iterative procedure is as follows:

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \mathbf{s}^k, \quad \text{where } B_k \mathbf{s}^k = -\mathbf{g}(\mathbf{x}^k), \quad \mathbf{x}^0 \text{ given.}$$

The process is a Newton method if $B_k = \mathbf{g}'(\mathbf{x}^k)$, and a Newton-like method if B_k is an approximation to $\mathbf{g}'(\mathbf{x}^k)$, see for instance [5, 9, 10].

*(zjbai@math.cuhk.edu.hk) Department of Mathematics, Chinese University of Hong Kong, Hong Kong, China.

The Newton and Newton-like Methods are attractive because of their rapid convergence from any sufficiently good initial guess $\mathbf{x}^0$. However, in each Newton or Newton-like iteration, we have to solve exactly the Jacobian or approximate Jacobian equation

$$B_k \mathbf{s}^k = -\mathbf{g}(\mathbf{x}^k). \tag{2}$$

Computing the exact solution using a direct method such as Gaussian elimination can be expensive if the number of unknowns is large and may not justified when $\mathbf{x}^k$ is far from a solution. Thus we can use an iterative method and solve (2) only approximately, i.e. by the inexact Newton or Newton-like method:

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \mathbf{s}^k, \quad \text{where } B_k \mathbf{s}^k = -\mathbf{g}(\mathbf{x}^k) + \mathbf{r}^k, \quad \|\mathbf{r}^k\|/\|\mathbf{g}(\mathbf{x}^k)\| \le \eta_k,$$

where $\|\cdot\|$ denotes the Euclidean vector norm or its corresponding induced matrix norm. In general, the nonnegative *forcing term* η_k is given in terms of $\mathbf{g}(\mathbf{x}^k)$, see for instance [4, 6, 7, 9].

When n is large, solving the Jacobian or approximate Jacobian equation will be costly. The cost can be reduced by using iterative methods (the inner iterations). However, if η_k is too small, an iterative method may *oversolve* the Jacobian or approximate Jacobian equation in the sense that the last tens or hundreds inner iterations before convergence may not improve the convergence of the outer Newton or Newton-like iterations. That is, additional accuracy in solving the Jacobian or approximate Jacobian equation requires additional expense, but results in little or no progress toward a solution [6].

In this paper, we discuss the use of inexact methods to two Newton-like methods, i.e. Method II and Method III in [8], for solving the IEP. Our inexact methods solve the approximate Jacobian equation inexactly by stopping the inner iterations before convergence, and thereby alleviate or minimize the oversolving problem. We will show that the convergence rates of our methods are superlinear. However, by stopping the inner iterations earlier, we can reduce the total cost of the inner-outer iterations.

This paper is organized as follows. In §2, we give some background knowledge about the Newton method for the IEP. Then we investigate the inexact method to Method II and Method III of [8] in §3–§4 respectively. We give the convergence analysis of the inexact methods with illustrative numerical tests.

2 The Newton Method

In this section, we briefly recall the Newton method for solving the IEP. For details, see [2, 8, 13]. For any $\mathbf{c} = (c_1, \ldots, c_n)^T \in \mathbb{R}^n$, define $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n$ by

$$\mathbf{f}(\mathbf{c}) \equiv (\lambda_1(\mathbf{c}) - \lambda_1^*, \ldots, \lambda_n(\mathbf{c}) - \lambda_n^*)^T = \mathbf{0}. \tag{3}$$

Clearly, $\mathbf{c}^*$ is a solution to the IEP if and only if $\mathbf{f}(\mathbf{c}^*) = \mathbf{0}$. Therefore, we can formulate the IEP as a system of nonlinear equations $\mathbf{f}(\mathbf{c}) = \mathbf{0}$.

As in [8], we assume that the given eigenvalues $\{\lambda_i^*\}_{i=1}^n$ are distinct. Then the eigenvalues of $A(\mathbf{c})$ are distinct too in some neighborhood of $\mathbf{c}^*$. It follows that the function $\mathbf{f}(\mathbf{c})$ is analytic in the same neighborhood and that the Jacobian of $\mathbf{f}$ is given by

$$\Big[J(\mathbf{c})\Big]_{ij} = \frac{\partial [\mathbf{f}(\mathbf{c})]_i}{\partial c_j} = \frac{\partial \lambda_i(\mathbf{c})}{\partial c_j} = \mathbf{q}_i(\mathbf{c})^T \frac{\partial A(\mathbf{c})}{\partial c_j} \mathbf{q}_i(\mathbf{c}), \quad 1 \le i, j \le n,$$

where $\mathbf{q}_i(\mathbf{c})$ are the normalized eigenvectors of $A(\mathbf{c})$ corresponding to the eigenvalues $\lambda_i(\mathbf{c})$, see [13, Eq. (4.6.2)] or [2, 3]. Hence by (1),

$$\Big[J(\mathbf{c})\Big]_{ij} = \mathbf{q}_i(\mathbf{c})^T A_j \mathbf{q}_i(\mathbf{c}), \quad 1 \le i, j \le n. \tag{4}$$

Thus by (1) again, we have

$$[J(\mathbf{c})\mathbf{c}]_i = \sum_{j=1}^{n} c_j \mathbf{q}_i(\mathbf{c})^T A_j \mathbf{q}_i(\mathbf{c}) = \mathbf{q}_i(\mathbf{c})^T A(\mathbf{c}) \mathbf{q}_i(\mathbf{c}) = \lambda_i(\mathbf{c}), \quad 1 \le i, j \le n,$$

i.e. $J(\mathbf{c})\mathbf{c} = (\lambda_1(\mathbf{c}), \cdots, \lambda_n(\mathbf{c}))^T$. By (3), this becomes

$$J(\mathbf{c})\mathbf{c} = \mathbf{f}(\mathbf{c}) + \boldsymbol{\lambda}^*, \tag{5}$$

where $\boldsymbol{\lambda}^* \equiv (\lambda_1^*, \cdots, \lambda_n^*)^T$.

Recall that the Newton method for $\mathbf{f}(\mathbf{c}) = \mathbf{0}$ is given by $J(\mathbf{c}^k)(\mathbf{c}^{k+1} - \mathbf{c}^k) = -\mathbf{f}(\mathbf{c}^k)$. By (5), this can be rewritten as

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

To summarize, we have the following Newton method for solving the IEP.

Algorithm 1: The Newton Method

For $k = 0$ until convergence, do:

1. *Compute the eigen-decomposition of $A(\mathbf{c}^k)$:*

$$Q(\mathbf{c}^k)^T A(\mathbf{c}^k) Q(\mathbf{c}^k) = \mathrm{diag}(\lambda_1(\mathbf{c}^k), \cdots, \lambda_n(\mathbf{c}^k)),$$

   where $Q(\mathbf{c}^k) = [\mathbf{q}_1(\mathbf{c}^k), \cdots, \mathbf{q}_n(\mathbf{c}^k)]$ is orthogonal.

2. *Form the Jacobian matrix: $[J(\mathbf{c}^k)]_{ij} = \mathbf{q}_i(\mathbf{c}^k)^T A_j \mathbf{q}_i(\mathbf{c}^k)$.*

3. *Solve $\mathbf{c}^{k+1}$ from the Jacobian equation: $J(\mathbf{c}^k)\mathbf{c}^{k+1} = \boldsymbol{\lambda}^*$.*

Notice that in Step 1, we have to compute all the eigenvalues and eigenvectors of $A(\mathbf{c}^k)$ exactly. This method converges Q-quadratically, see for instance [8, Theorem 3.2] and [13, Theorem 4.6.1]. Here, we recall the definition of two kind of convergence rates, see [4] and [10, Chap. 9].

Definition 1 *Let $\{\mathbf{x}^k\}$ be a sequence which converges to $\mathbf{x}^*$. Then*

1. *$\mathbf{x}^k \to \mathbf{x}^*$ with Q-convergence rate at least q $(q > 1)$ if*

$$\|\mathbf{x}^{k+1} - \mathbf{x}^*\| = O(\|\mathbf{x}^k - \mathbf{x}^*\|^q) \quad as\, k \to \infty; \tag{6}$$

2. *$\mathbf{x}^k \to \mathbf{x}^*$ with R-convergence rate at least q $(q > 1)$ if*

$$\limsup_{k \to \infty} \|\mathbf{x}^k - \mathbf{x}^*\|^{1/q^k} < 1 \quad as\, k \to \infty. \tag{7}$$

3 The Inexact Newton-like Method

In this section, we first recall Method II in [8], and then give the inexact version.

3.1 The Inexact Newton-like Method

In Algorithm 1 all the eigenvectors of $A(\mathbf{c})$ have to be computed exactly per step, which is very time consuming. Therefore, we consider approximating these eigenvectors. Suppose that we have determined an estimate $\mathbf{c}^k$ of $\mathbf{c}^*$ and an approximation $Q_k = [\mathbf{q}_1^k, \ldots, \mathbf{q}_n^k]$ to the orthogonal matrix of eigenvectors $Q(\mathbf{c}^k)$. Then we compute a new estimate $\mathbf{c}^{k+1}$ by solving

$$J_k \mathbf{c}^{k+1} = \boldsymbol{\lambda}^*, \tag{8}$$

where

$$J_k = [(\mathbf{q}_i^k)^T A_j \mathbf{q}_i^k].$$

To update our approximations to the eigenvectors, we apply one step inverse iteration; that is, we compute $\mathbf{w}_i$, $i = 1, \ldots, n$ by solving

$$(A(\mathbf{c}^{k+1}) - \lambda_i I)\mathbf{w}_i = \mathbf{q}_i^k, \quad i = 1, \ldots, n. \tag{9}$$

We then define $Q_{k+1} = [\mathbf{q}_1^{k+1}, \ldots, \mathbf{q}_n^{k+1}]$ by

$$\mathbf{q}_i^{k+1} = \frac{\mathbf{w}_i}{\|\mathbf{w}_i\|}, \quad i = 1, \ldots, n. \tag{10}$$

It is showed that Method II is Q-quadratically convergent under the assumption that the systems (9) and (8) are solved exactly, see [8, 3] and [13, Theorem 4.6.2]. In Method II, there are two inner iterations: the one-step inverse power method (9) and the approximate Jacobian equation (8). In order to avoid the *oversolving* of the inner iterations, we have to look for suitable tolerances small enough to guarantee the convergence of the outer iterations, but large enough to reduce the oversolving problem of the inner iterations.

Nonetheless, we find that the tolerance for the inverse power method (9) can be set very large (to 1/4), whereas the tolerance for the approximate Jacobian equation (8) has to be small in order to have a superlinear convergence for the outer iteration. The tolerance for (8) will be in terms of a computable quantity in our inexact algorithm. Below we give our algorithm.

Algorithm 2: The Inexact Newton-Like Method

1. *Given* $\mathbf{c}^0$, *iterate Algorithm 1 once to obtain* $\mathbf{c}^1$ *and write* $P_0 = [\mathbf{p}_1^0, \cdots, \mathbf{p}_n^0] = [\mathbf{q}_1(\mathbf{c}^0) \cdots, \mathbf{q}_n(\mathbf{c}^0)]$.

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

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

   $$(A(\mathbf{c}^k) - \lambda_i^* I)\mathbf{w}_i^k = \mathbf{p}_i^{k-1} + \mathbf{t}_i^k, \quad 1 \le i \le n, \tag{11}$$

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

   $$\|\mathbf{t}_i^k\| \le \frac{1}{4}. \tag{12}$$

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

$$\mathbf{p}_i^k = \frac{\mathbf{w}_i^k}{\|\mathbf{w}_i^k\|}, \quad 1 \le i \le n. \tag{13}$$

(c) Form the approximate Jacobian matrix:

$$[J_k]_{ij} = (\mathbf{p}_i^k)^T A_j \mathbf{p}_i^k, \quad 1 \le i, j \le n. \tag{14}$$

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

$$J_k \mathbf{c}^{k+1} = \boldsymbol{\lambda}^* + \mathbf{r}^k, \tag{15}$$

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

$$\|\mathbf{r}^k\| \le \Big(\max_{1 \le i \le n} \frac{1}{\|\mathbf{v}_i^k\|} \Big)^{\beta}, \qquad 1 < \beta \le 2. \tag{16}$$

Note that the main difference between Algorithm 2 and Method II is that we solve (11) and (15) approximately rather than exactly as in (9) and (8).

3.2 Rate of Convergence

In this subsection, we will show that the Q-convergence rate of Algorithm 2 is β.

As in [8], we assume that the given eigenvalues $\{\lambda_i^*\}_{i=1}^n$ are distinct and that the Jacobian $J(\mathbf{c}^*)$ defined in (4) is nonsingular. Then we have the following theorem on the rate of convergence.

Theorem 1 [2] *Let the given eigenvalues* $\{\lambda_i^*\}_{i=1}^n$ *be distinct and the Jacobian* $J(\mathbf{c}^*)$ *be nonsingular. Then Algorithm 2 is locally convergent with Q-convergence rate* β*.*

For numerical examples, we refer to [2].

4 The Inexact Cayley Transform Method

In this section, we briefly recall Method III in [8], and then give the inexact version, i.e. the inexact Cayley transform method.

4.1 The Inexact Cayley Transform Method

Method III in [8] is based on Cayley transforms. A solution to the IEP can be described by $\mathbf{c}$ and U, where U is an orthogonal matrix and

$$U^T A(\mathbf{c}) U = \Lambda_*, \quad \Lambda_* = \mathrm{diag}(\lambda_1^*, \ldots, \lambda_n^*). \tag{17}$$

Suppose that an orthogonal matrix U_k is the current approximations of U. Define $e^{Z_k} \equiv U_k^T U$. Then Z_k is a skew-symmetric matrix and (17) can be written as

$$U_k^T A(\mathbf{c}) U_k = e^{Z_k} \Lambda_* e^{-Z_k} = \Lambda_* + Z_k \Lambda_* - \Lambda_* Z_k + O(\|Z_k\|^2). \tag{18}$$

In Method III, we define a new estimate of $\mathbf{c}^{k+1}$ by neglecting the second order terms in Z_k, and equating the diagonal elements in (18), i.e. $\mathbf{c}^{k+1}$ is given by

$$(\mathbf{u}_i^k)^T A(\mathbf{c}^{k+1})\mathbf{u}_i^k = \lambda_i^*, \quad i = 1, \ldots, n, \tag{19}$$

where $\{\mathbf{u}_i^k\}_{i=1}^n$ are the column vectors of U_k. By (1), (19) can be rewritten as

$$J_k \mathbf{c}^{k+1} = \boldsymbol{\lambda}^*, \tag{20}$$

where $\boldsymbol{\lambda}^* \equiv (\lambda_1^*, \ldots, \lambda_n^*)^T$ and J_k is the approximate Jacobian matrix with entries

$$[J_k]_{ij} = (\mathbf{u}_i^k)^T A_j \mathbf{u}_i^k, \qquad i, j = 1, \ldots, n. \tag{21}$$

Once we get $\mathbf{c}^{k+1}$ from (20), we obtain Z_k by neglecting the second order terms in Z_k, and equating the off-diagonal elements in (18), i.e.

$$[Z_k]_{ij} = \frac{(\mathbf{u}_i^k)^T A(\mathbf{c}^{k+1})\mathbf{u}_j^k}{\lambda_j^* - \lambda_i^*}, \qquad 1 \le i \ne j \le n. \tag{22}$$

Finally we update U_k by setting $U_{k+1} = U_k S_k$, where S_k is an orthogonal matrix constructed by the Cayley transform

$$S_k = (I + \frac{1}{2} Z_k)(I - \frac{1}{2} Z_k)^{-1}.$$

The Q-quadratic convergence of Method III was proven under the assumption that the approximated Jacobian equation (20) is solved exactly, see [8]. To reduce the *oversolving* of the inner iterations, we have to look for suitable tolerances to improve the efficiency.

For the nonlinear system $\mathbf{f}(\mathbf{c}) = \mathbf{0}$, the stopping criterion is usually given in terms of $\mathbf{f}(\mathbf{c})$. By (3), this will involve computing $\lambda_i(\mathbf{c}^k)$ of $A(\mathbf{c}^k)$ which are costly to compute. Our idea is to replace them by Rayleigh quotients, see (25) and (27) below. Thus we have the following algorithm.

Algorithm 3: Inexact Cayley Transform Method

1. *Given* $\mathbf{c}^0$, *compute the orthonormal eigenvectors* $\{\mathbf{q}_i(\mathbf{c}^0)\}_{i=1}^n$ *and the eigenvalues* $\{\lambda_i(\mathbf{c}^0)\}_{i=1}^n$ *of* $A(\mathbf{c}^0)$. *Let* $V_0 = [\mathbf{v}_1^0, \ldots, \mathbf{v}_n^0] = [\mathbf{q}_1(\mathbf{c}^0), \ldots, \mathbf{q}_n(\mathbf{c}^0)]$, *and*

$$\boldsymbol{\rho}^0 = (\lambda_1(\mathbf{c}^0), \ldots, \lambda_n(\mathbf{c}^0))^T.$$

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

   (a) *Form the approximate Jacobian matrix:*

$$[J_k]_{ij} = (\mathbf{v}_i^k)^T A_j \mathbf{v}_i^k, \quad 1 \le i, j \le n. \tag{23}$$

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

$$J_k \mathbf{c}^{k+1} = \boldsymbol{\lambda}^* + \mathbf{r}^k, \tag{24}$$

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

$$\|\mathbf{r}^k\| \le \|\boldsymbol{\rho}^k - \boldsymbol{\lambda}^*\|^\beta, \quad \beta \in (1, 2]. \tag{25}$$

(c) Form the skew-symmetric matrix Y_k:

$$[Y_k]_{i\,j} = \frac{(\mathbf{v}_i^k)^T\, A(\mathbf{c}^{k+1})\, \mathbf{v}_j^k}{\lambda_j^* - \lambda_i^*}, \quad 1 \le i \ne j \le n.$$

(d) Compute $V_{k+1} = [\mathbf{v}_1^{k+1}, \ldots, \mathbf{v}_n^{k+1}]$ by solving

$$(I + \frac{1}{2}Y_k)V_{k+1}^T = (I - \frac{1}{2}Y_k)V_k^T. \tag{26}$$

(e) Compute $\boldsymbol{\rho}^{k+1} = (\rho_1^{k+1}, \ldots, \rho_n^{k+1})^T$ by

$$\rho_i^{k+1} = (\mathbf{v}_i^{k+1})^T\, A(\mathbf{c}^{k+1})\, \mathbf{v}_i^{k+1}, \quad i = 1, \ldots, n. \tag{27}$$

4.2 Rate of Convergence

In the following, we show that the R-convergence rate of Algorithm 3 is at least β.

As in [8], we assume that the given eigenvalues $\{\lambda_i^*\}_{i=1}^n$ are distinct and the Jacobian $J(\mathbf{c}^*)$ is nonsingular. Then we have the following results on the rate of convergence.

Theorem 2 [1] *Let the given eigenvalues $\{\lambda_i^*\}_{i=1}^n$ be distinct and $J(\mathbf{c}^*)$ be nonsingular. Then the iterates $\{\mathbf{c}^k\}$ generated by Algorithm 3 converges to $\mathbf{c}^*$ with R-convergence rate at least equal to β.*

4.3 Numerical Experiments

In this subsection, we compare the numerical performance of Algorithm 3 with that of Method III. Our aim is to illustrate the advantage of our method over Method III in terms of the overall computational cost. We use Toeplitz matrices as our A_i in (1):

$$A_1 = I, A_2 = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 1 & 0 & 1 & \ddots & \vdots \\ 0 & 1 & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 0 & 1 \\ 0 & \cdots & 0 & 1 & 0 \end{bmatrix}, \cdots, A_n = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 0 & \ddots & \ddots & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \ddots & \ddots & 0 \\ 1 & 0 & \cdots & 0 & 0 \end{bmatrix}.$$

In particular, $A(\mathbf{c})$ is symmetric Toeplitz matrix having $\mathbf{c}$ as its first column. For demonstration purposes, we consider two different problem sizes: $n = 100$ and 200. For both n, the exact solutions $\mathbf{c}^*$ are chosen randomly so that the exact Jacobian $J(\mathbf{c}^*)$ is nonsingular. Then we compute the eigenvalues $\{\lambda_i^*\}_{i=1}^n$ of $A(\mathbf{c}^*)$ as the prescribed eigenvalues. Since both algorithms are locally convergent only, the initial guess $\mathbf{c}^0$ is chosen reasonably close to the true solution $\mathbf{c}^*$.

The linear systems (20) and (24) are both solved by the Matlab-provided QMR method [11, pp. 210–214]. We use $\mathbf{c}^k$, the iterant at the kth iteration, as the initial guess for the QMR method in the $(k+1)$th iteration. Moreover, we use 10^{-13} as the stopping tolerance for (20), and the stopping criterion for (24) is given in (25). The outer iterations of Method III and Algorithms 3 are stopped when

$$\|U_k^T A(\mathbf{c}^k)U_k - \Lambda_*\|_F \le 10^{-10} \quad \text{and} \quad \|V_k^T A(\mathbf{c}^k)V_k - \Lambda_*\|_F \le 10^{-10}$$

respectively.

In Tables 1–2, we give the total numbers of outer iterations N_o averaged over ten tests for different values of β. We also list the average numbers of inner iterations N_i required for solving the approximate Jacobian equations (20) in Method III and (24) in Algorithm 3.

	Method III	Alg. 3									
β		1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0
N_o	3.2	12	5.2	4	3.3	3.2	3.2	3.2	3.2	3.2	3.2
N_i	397	755	445	379	327	323	325	329	336	339	349

Table 1 : Averaged numbers of outer and inner iterations with dimension $n = 100$.

	Method III	Alg. 3									
β		1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0
N_o	3	10.9	6	4	3	3	3	3	3	3	3
N_i	818	1444	1144	855	684	719	725	732	738	747	763

Table 2 : Averaged numbers of outer and inner iterations with dimension $n = 200$.

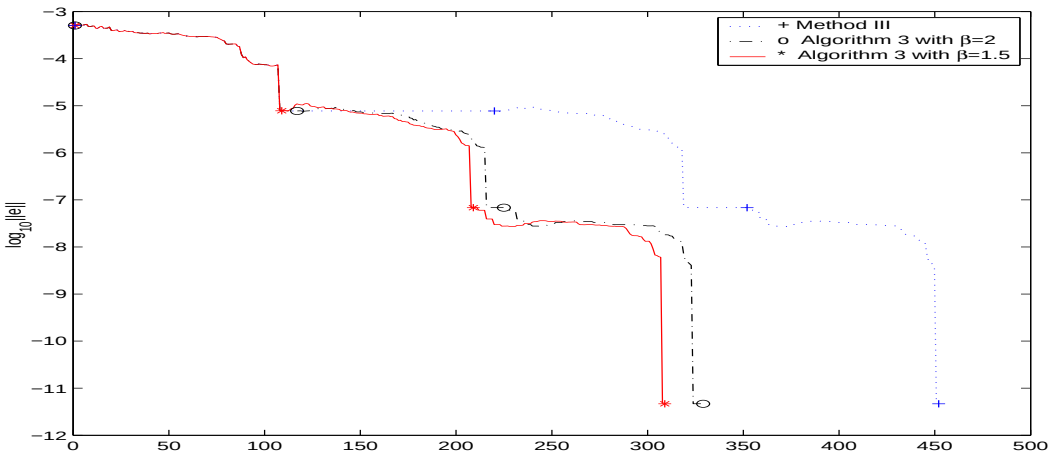

Figure 1: Convergence history of one of the test matrices with $n = 100$.

We observe from the tables that, in terms of the total number of outer iterations, for β between 1.5 and 2.0, our method converges at the same rate as Method III. However, in terms of the total numbers of inner iterations, we note that for β between 1.4 and 2.0, Algorithm 3 is in fact more effective than Method III. The most effective β is between 1.4 and 1.5.

To further illustrate the oversolving problem, we investigate the convergence history of the approximations to $\mathbf{c}^*$ at the inner iterations. Specifically, at each inner iteration, we computed the 2-norm error e between the current approximation and $\mathbf{c}^*$. Figure 1 depicts the logarithm of e versus the number of inner iterations for one of the test matrices of dimension $n = 100$ solved by Method III and Algorithm 3 with $\beta = 1.5$ and 2. We mark the error at the outer iterations with special symbols too. We can see that our method converges faster than Method III. Moreover, for Method III, oversolving problem is significant (see

the horizontal lines between iteration numbers 100 to 200, and 325 to 350) whereas there is no oversolving for Algorithm 3 with $\beta = 1.5$.

References

[1] Z. Bai, R. Chan, and B. Morini, *An Inexact Cayley Transform Method for Inverse Eigenvalue Problems*, submitted to Inverse Problems.

[2] R. Chan, H. Chung, and S. Xu, *The Inexact Newton-Like Method for Inverse Eigenvalue Problem*, accepted for publication by BIT.

[3] R. Chan, S. Xu, and H. Zhou, *On the Convergence Rate of a Quasi-Newton Method for Inverse Eigenvalue Problem*, *SIAM J. Numer. Anal.*, **36** (1999), 436–441.

[4] R. Dembo, S. Eisenstat, and T. Steihaug, *Inexact Newton Methods*, *SIAM J. Numer. Anal.*, **19** (1982), 400–408.

[5] J. Dennis and R. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice Hall, Englewood Cliff, NJ, 1983.

[6] S. Eisenstat and H. Walker, *Choosing the Forcing Terms in an Inexact Newton Method*, *SIAM J. Sci. Comput.*, **17** (1996), 16–32.

[7] D. Fokkema, G. Sleijen, and H. Vorst, *Accelerated Inexact Newton Schemes for Large Systems of Nonlinear Equations*, *SIAM J. Sci. Comput.*, **19** (1998), 657–674.

[8] S. Friedland, J. Nocedal, and M. Overton, *The Formulation and Analysis of Numerical Methods for Inverse Eigenvalue Problems*, *SIAM J. Numer. Anal.*, **24** (1987), 634–667.

[9] B. Morini, *Convergence Behaviour of Inexact Newton Methods*, *Math. Comput.*, **68** (1999), 1605–1613.

[10] J. Ortega and W. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, 1970.

[11] Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS Publishing Company, 1996.

[12] J. Sun, *Backward Errors for the Inverse Eigenvalue Problem*, *Numer. Math.*, **82** (1999), 339–349.

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